

February 25, 2005

Candidate Standard 5104.2-2005



The Printer Working Group

# Print Service Interface Version 1.0

Status: Approved

**Abstract:** The Print Service Interface is the set of interfaces and methods that enable a Client, such as a Printer, Mobile Device, Web Portal, or Service, to create a Print Job on a Print Service. The Print Service may be used to resolve and access the information to be printed.

This document is a PWG Candidate Standard. For a definition of a "PWG Candidate Standard", see: <ftp://ftp.pwg.org/pub/pwg/general/pwg-process20.pdf>

This document is available electronically at:

<ftp://ftp.pwg.org/pub/pwg/candidates/cs-psi10-20050225-5104.2.pdf>, .doc

**Copyright (C) 2002-2005, The Printer Working Group. All rights reserved.**

This document may be copied and furnished to others, and derivative works that comment on, or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice, this paragraph and the title of the Document as referenced below are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Printer Working Group, a program of the IEEE-ISTO.

Title: PWG Print Service Interface Version 1.0

The IEEE-ISTO and the Printer Working Group DISCLAIM ANY AND ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED INCLUDING (WITHOUT LIMITATION) ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

The Printer Working Group, a program of the IEEE-ISTO, reserves the right to make changes to the document without further notice. The document may be updated, replaced or made obsolete by other documents at any time.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO take no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights.

The IEEE-ISTO and the Printer Working Group, a program of the IEEE-ISTO invite any interested party to bring to its attention any copyrights, patents, or patent applications, or other proprietary rights, which may cover technology that may be required to implement the contents of this document. The IEEE-ISTO and its programs shall not be responsible for identifying patents for which a license may be required by a document and/or IEEE-ISTO Industry Group Standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention. Inquiries may be submitted to the IEEE-ISTO by e-mail at:

[info@ieee-isto.org](mailto:info@ieee-isto.org)

The Printer Working Group acknowledges that the IEEE-ISTO (acting itself or through its designees) is, and shall at all times, be the sole entity that may authorize the use of certification marks, trademarks, or other special designations to indicate compliance with these materials.

Use of this document is wholly voluntary. The existence of this document does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to its scope.

## About the IEEE-ISTO

The IEEE-ISTO is a not-for-profit corporation offering industry groups an innovative and flexible operational forum and support services. The IEEE Industry Standards and Technology Organization member organizations include printer manufacturers, print server developers, operating system providers, network operating systems providers, network connectivity vendors, and print management application developers. The IEEE-ISTO provides a forum not only to develop standards, but also to facilitate activities that support the implementation and acceptance of standards in the marketplace. The organization is affiliated with the IEEE (<http://www.ieee.org/>) and the IEEE Standards Association (<http://standards.ieee.org/>).

For additional information regarding the IEEE-ISTO and its industry programs visit:  
<http://www.ieee-isto.org>.

## About the Printer Working Group

The Printer Working Group (or PWG) is a Program of the IEEE-ISTO. All references to the PWG in this document implicitly mean "The Printer Working Group, a Program of the IEEE ISTO." The PWG is chartered to make printers and the applications and operating systems supporting them work together better. In order to meet this objective, the PWG will document the results of their work as open standards that define print related protocols, interfaces, data models, procedures and conventions. Printer manufacturers and vendors of printer related software would benefit from the interoperability provided by voluntary conformance to these standards.

In general, a PWG standard is a specification that is stable, well understood, and is technically competent, has multiple, independent and interoperable implementations with substantial operational experience, and enjoys significant public support.

## Contact information:

The Printer Working Group  
c/o The IEEE Industry Standards and Technology Organization  
445 Hoes Lane  
Piscataway, NJ 08854  
USA

PSI Web Page: <http://www.pwg.org/ps>      PSI Mailing List: [ps@pwg.org](mailto:ps@pwg.org)

Instructions for subscribing to the PSI mailing list can be found at the following link:  
<http://www.pwg.org/mailhelp.html>

Members of the PWG and interested parties are encouraged to join the PWG and PSI WG mailing lists in order to participate in discussions, clarifications and review of the WG product. Requests for additional media names for inclusion in this specification should be sent to the PSI mailing list for consideration.

## Contributors

Lee Farrell	Canon
Alan Berkema	Hewlett-Packard Company
Bob Taylor	Hewlett-Packard Company
Dave Hall	Hewlett-Packard Company
Ira McDonald	High North
Harry Lewis	IBM
Dennis Carney	IBM
Don Wright	Lexmark
Jerry Thrasher	Lexmark
Don Levinstone	Motorola
Bill Wagner	NetSilicon
Ted Tronson	Novell
Gail Songer	Peerless
Kirk Ocke	Xerox
Peter Zehler	Xerox

# 1 Contents

## 1.1 Table of Contents

<b>1</b>	<b>Contents .....</b>	<b>5</b>
1.1	Table of Contents.....	5
1.2	Table of Interfaces and Methods .....	10
1.3	Table of Figures.....	12
<b>2</b>	<b>Terminology .....</b>	<b>13</b>
<b>3</b>	<b>Print Service Interface Protocol.....</b>	<b>14</b>
3.1	PSI Protocol.....	14
3.2	PSI Protocol Stack.....	15
<b>4</b>	<b>Interface Definition.....</b>	<b>16</b>
4.1	PSI Service Root URL.....	18
4.2	Negotiating a secure PSI connection .....	18
4.3	QueryEndpointsInterface.....	19
4.3.1	Interface Usage Examples .....	20
4.3.2	QuerySupportedInterfaces .....	20
4.3.3	QueryInterfaceDefinition.....	21
4.4	ServiceCapabilitiesInterface .....	22
4.4.1	Interface Usage Examples .....	22
4.4.2	GetTargetDeviceElements.....	23
4.4.3	GetKnownTargetDevices .....	24
4.4.4	ValidateReference.....	26
4.5	JobControlInterface .....	28
4.5.1	Interface Usage Examples .....	29
4.5.2	CreateJob .....	31
4.5.3	CloseJob.....	33
4.5.4	AddDocumentByReference.....	34
4.5.5	AddDocumentByPush .....	36
4.5.6	PushDocumentDataDelivered.....	37
4.5.7	AddDocumentByValue.....	39
4.5.8	GetJobs .....	40
4.5.9	GetJobElements .....	42
4.5.10	SetJobElements.....	43

- 4.5.11 CancelJob..... 44
- 4.5.12 GetDocuments ..... 45
- 4.5.13 GetDocumentElements ..... 46
- 4.5.14 SetDocumentElements..... 47
- 4.5.15 CancelDocument..... 48
- 4.5.16 FetchDocumentDataByPull ..... 49
- 4.5.17 PullDocumentDataFetched ..... 50
- 4.5.18 FetchDocumentDataByValue ..... 51
- 4.5.19 FetchJobs ..... 52
- 4.6 TargetDeviceSupportInterface..... 54
  - 4.6.1 Interface Usage Examples ..... 55
  - 4.6.2 FetchNextJob ..... 56
  - 4.6.3 CloseJob..... 58
  - 4.6.4 FetchNextDocumentByPull..... 59
  - 4.6.5 PullDocumentDataFetched..... 60
  - 4.6.6 FetchNextDocumentByValue..... 61
  - 4.6.7 SendJobNotification ..... 62
  - 4.6.8 SendDocumentNotification ..... 63
  - 4.6.9 RegisterTargetDevice ..... 65
  - 4.6.10 UnregisterTargetDevice..... 66
  - 4.6.11 SendTargetDeviceNotification ..... 67
- 5 Print Job Flows enabled by PSI ..... 69**
  - 5.1 Print Service is configured for target device specification and does not auto select:..... 69
  - 5.2 Print Service is not configured for target device specification and always auto selects: ..... 70
  - 5.3 Print Service is configured for target device specification and auto selection:..... 70
- 6 Type Definitions ..... 72**
  - 6.1 PWG Standard URI Query Parameters..... 72
    - 6.1.1 'attid'..... 72
    - 6.1.2 'auth' ..... 72
    - 6.1.3 'binary'..... 72
    - 6.1.4 'cert' ..... 73
    - 6.1.5 'domain'..... 73
    - 6.1.6 'legacy' ..... 73
    - 6.1.7 'location'..... 73
    - 6.1.8 'msgid' ..... 73
    - 6.1.9 'name' ..... 73
    - 6.1.10 'passive' ..... 73
    - 6.1.11 'pw'..... 74
    - 6.1.12 'sec'..... 74
    - 6.1.13 'user' ..... 74
  - 6.2 Target Device Identifier..... 74

6.2.1	raw-tcp .....	75
6.2.2	fax .....	75
6.2.3	pwg-psips, pwg-psitd .....	75
6.2.4	ipp .....	76
6.2.5	ippfax .....	76
6.2.6	lpr .....	76
6.3	Reference .....	77
6.3.1	http .....	77
6.3.2	ftp .....	77
6.3.3	file .....	78
6.3.4	pop .....	78
6.3.5	imap .....	79
6.4	InterfaceIdentifier .....	81
<b>7</b>	<b>Conformance .....</b>	<b>83</b>
7.1	PSI/1.0 Common Conformance .....	83
7.1.1	PSI/1.0 Common Implementation Requirements .....	83
7.1.2	PSI/1.0 Common Implementation Recommendations .....	83
7.1.3	PSI/1.0 Common Literature Recommendations .....	84
7.2	PSI/1.0 Client Conformance .....	84
7.2.1	PSI/1.0 Client Requirements .....	84
7.2.2	PSI/1.0 Client Recommendations .....	85
7.3	PSI/1.0 Print Service Conformance .....	85
7.3.1	PSI/1.0 Print Service Requirements .....	85
7.3.2	PSI/1.0 Print Service Recommendations .....	85
7.4	PSI/1.0 Target Device Conformance .....	86
7.4.1	PSI/1.0 Target Device Requirements .....	86
7.4.2	PSI/1.0 Target Device Recommendations .....	86
7.5	PSI/1.0 Method Support Requirements .....	87
7.6	Printer Object Element Support Requirements .....	88
7.6.1	PrinterDescription .....	88
7.6.2	PrinterStatus .....	88
7.6.3	ProcessingDefaults, ProcessingReady, ProcessingSupported .....	88
7.7	Job Object Element Support Requirements .....	88
7.7.1	JobStatus Elements .....	88
7.7.2	JobDescription Elements .....	89
7.7.3	JobProcessing Elements .....	89
7.8	Document Object Element Support Requirements .....	89
7.8.1	DocumentStatus Elements .....	89
7.8.2	DocumentDescription Elements .....	89
7.8.3	DocumentProcessing Elements .....	89

7.9 Job Lifetime ..... 89

**8 Normative References ..... 90**

**9 Informative References ..... 91**

**10 PWG and IANA Considerations ..... 92**

**11 Internationalization Considerations ..... 93**

**12 Security Model and Considerations ..... 94**

12.1 Internet Threat Model ..... 94

12.1.1 Passive Attacks ..... 95

12.1.2 Active Attacks ..... 95

12.2 Enterprise Threat Model ..... 96

12.3 Mobile Threat Model ..... 97

12.4 HTTP Threat Model ..... 97

**13 Acronyms ..... 98**

**14 Appendix A – Legacy URI schemes (Informative) ..... 99**

14.1 Target Device Identifier ..... 99

14.1.1 smb ..... 99

14.2 Reference ..... 99

14.2.1 smb: ..... 99

**15 Appendix B – Example XML Instance Objects (Informative) ..... 100**

15.1 unsupportedElements : UnsupportedElements ..... 100

15.2 printerFilterElements : Printer ..... 100

15.3 jobFilterElements : Job ..... 100

15.4 documentFilterElements : Document ..... 101

**16 Appendix C – Standard Print System Events (Normative) ..... 102**

16.1 Introduction ..... 102

16.1.1 Rationale for Defining Standard Events ..... 102

16.1.2 Informative References in Standard Events ..... 102

16.1.3 Naming of Document Events ..... 102

16.1.4 Naming of Job Events ..... 102

16.1.5 Naming of Printer Events ..... 102



- 16.1.6 Naming of Subunit Events..... 103
- 16.2 Terminology ..... 103
  - 16.2.1 Model Terminology..... 103
  - 16.2.2 Conformance Terminology..... 103
- 16.3 Standard Document Events..... 104
  - 16.3.1 DocumentStateChanged ..... 104
  - 16.3.2 DocumentConfigChanged ..... 104
  - 16.3.3 DocumentProgress..... 104
- 16.4 Standard Job Events..... 105
  - 16.4.1 JobStateChanged..... 105
  - 16.4.2 JobConfigChanged ..... 106
  - 16.4.3 JobProgress ..... 106
- 16.5 Standard Printer Events ..... 106
  - 16.5.1 PrinterStateChanged ..... 106
  - 16.5.2 PrinterConfigChanged ..... 107
  - 16.5.3 PrinterQueueOrderChanged ..... 108
- 16.6 Standard Subunit Events..... 108
  - 16.6.1 Alert Management ..... 108
  - 16.6.2 General (Print System) ..... 108
  - 16.6.3 Input Tray ..... 108
  - 16.6.4 Interpreter ..... 109
  - 16.6.5 Marker..... 109
  - 16.6.6 Marker Supplies..... 109
  - 16.6.7 Media Path..... 109
  - 16.6.8 Output Bin ..... 109
  - 16.6.9 Subunit (Generic)..... 110
- 16.7 Vendor Events ..... 110
- 16.8 Best Practices for Future Events Standards..... 111
- 16.9 Normative References (for this appendix)..... 111
- 16.10 Informative References (for this appendix)..... 111

## 1.2 Table of Interfaces and Methods

QuerySupportedInterfaces () : interfaceIdentifiers : InterfaceIdentifierContainer .....	20
QueryInterfaceDefinition (interfaceIdentifier : InterfaceIdentifier) : interfaceEndPointURI : URI, interfaceWSDLURL : URI .....	21
GetTargetDeviceElements (targetDeviceIdentifier : URI, requestedPrinterElements : RequestedElements) : printer : Printer, unsupportedElements : UnsupportedElements .....	23
GetKnownTargetDevices (targetDeviceIdentifierFilter : URI, printerFilterElements : Printer) : targetDeviceIdentifiers : TargetDeviceIdentifierContainer, unsupportedElements : UnsupportedElements .....	24
ValidateReference (reference : URI) : validReference : Boolean .....	26
<b>Figure 10 Print Service to Target Device by Reference .....</b>	<b>30</b>
CreateJob (targetDeviceIdentifier : URI, autoSelectTargetDevice : Boolean, requestedTargetDeviceDataType : DocumentFormatDetails, jobDescription : JobDescription, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing) : jobURI : URI, unsupportedElements : UnsupportedElements, targetDeviceIdentifierAutoSelect : Boolean; .....	31
CloseJob (jobURI : URI) : void.....	33
AddDocumentByReference (jobURI : URI, reference : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing) : documentNumber : int, unsupportedElements : UnsupportedElements .....	34
AddDocumentByPush (jobURI : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing) : documentNumber : int, dataSinkURIs : DataSinkContainer, unsupportedElements : UnsupportedElements .....	36
PushDocumentDataDelivered (jobURI : URI, documentNumber : int, dataSinkURI : URI, documentOctetsSent : long) : void .....	37
AddDocumentByValue (jobURI : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, documentData : base64Binary) : documentNumber : int, unsupportedElements : UnsupportedElements .....	39
GetJobs (jobFilterElements : Job, requestedJobElements : RequestedElements, limit : int) : jobs : JobContainer, callbackInterval : uint32, unsupportedElements : UnsupportedElements.....	40
GetJobElements (jobURI : URI, requestedJobElements : RequestedElements) : job : Job, callbackInterval : uint32, unsupportedElements : UnsupportedElements .....	42
SetJobElements (jobURI : URI, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing) : unsupportedElements : UnsupportedElements.....	43
CancelJob (jobURI : URI, documentMessage : String) : void .....	44

GetDocuments (jobURI : URI, documentFilterElements : Document, requestedDocumentElements : RequestedElements) : documents : DocumentContainer, callbackInterval: uint32, unsupportedElements : UnsupportedElements..... 45

GetDocumentElements (jobURI : URI, documentNumber : int, requestedDocumentElements : RequestedElements) : document : Document, callbackInterval : uint32, unsupportedElements : UnsupportedElements..... 46

SetDocumentElements (jobURI : URI, documentNumber : int, documentProcessing : DocumentProcessing) : unsupportedElements : UnsupportedElements..... 47

CancelDocument (jobURI : URI, documentNumber : int, documentMessage : String) : void..... 48

FetchDocumentDataByPull (jobURI : URI, documentNumber : int) : dataSourceURIs : DataSourceContainer..... 49

PullDocumentDataFetched (jobURI : URI, documentNumber : int, dataSourceURI : URI) : void ..... 50

FetchDocumentDataByValue (jobURI : URI, documentNumber : int) : documentData : base64Binary. 51

FetchJobs (printService : URI, jobFilter : Job) : unsupportedElements : UnsupportedElements ..... 52

**TargetDeviceSupportInterface..... 54**

FetchNextJob(jobFilter : Job, targetDeviceIdentifier : URI) : jobURI : URI, sendJobNotifications : JobEvents, jobDescription : JobDescription, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing, callbackInterval : uint32, unsupportedElements : UnsupportedElements..... 56

CloseJob (jobURI : URI) : void..... 58

FetchNextDocumentByPull (jobURI : URI) : documentNumber : int, dataSourceURIs : DataSourceContainer, sendDocumentNotifications : DocumentEvents, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, lastDocument : Boolean, callbackInterval : uint32 ..... 59

PullDocumentDataFetched (jobURI : URI, documentNumber : int, dataSourceURI : URI) : void ..... 60

FetchNextDocumentByValue (jobURI : URI) : documentNumber : int, sendDocumentNotifications : DocumentEvents, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, lastDocument : Boolean, documentData : base64Binary, callbackInterval : uint32 ..... 61

SendJobNotification (jobURI : URI, targetDeviceIdentifier : URI, jobStatus : JobStatus, jobEvents : JobEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements..... 62

SendDocumentNotification (jobURI : URI, documentNumber : int, targetDeviceIdentifier : URI, documentStatus : DocumentStatus, documentEvents : DocumentEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements ..... 63

RegisterTargetDevice (targetDeviceIdentifier : URI, targetDeviceElements : Printer, registrationLifetimeRequest : int, deviceWillPoll : Boolean) : registrationLifetimeGranted : int, notifyEvents : NotifyEvents, unsupportedElements : UnsupportedElements ..... 65

UnregisterTargetDevice (targetDeviceIdentifier : URI) : void ..... 66

SendTargetDeviceNotification (targetDeviceIdentifier : URI, printerStatus : PrinterStatus, printerEvents : PrinterEvents, subunitEvents : SubunitEvents, vendorEvents : VendorEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements ..... 67

### 1.3 Table of Figures

Figure 1 Protocol Connection Diagram.....	14
Figure 2 PSI Protocol Stack.....	15
Figure 3 Interface Instantiation.....	17
Figure 4 PSI Service Root URL Paths.....	18
Figure 5 Bind to JobControlInterface.....	20
Figure 6 Client Finds Local Color Printers.....	22
Figure 7 Client Determines a printer / service capabilities.....	23
Figure 8 Print Service posts data to Target Device .....	29
Figure 9 PSI Client initiates a Reference Job .....	30
Figure 10 Print Service to Target Device by Reference .....	30
Figure 11 Target Device Polls .....	55
Figure 12 Client Informs Target Device of Job.....	56
Figure 13 Security Model and Considerations .....	94

## 2 Terminology

**Service** - A *service* provides some desired functions and contains one or more interfaces used for communication. A Print Service is an example of a *service*.

**Interface** - An *interface* is a collection of *methods* that are exposed by the service. An example of an *interface* is the Print Service JobControllInterface.

**Method** - A *method* is an operation in an interface.

**Protocol** - A *protocol* is an agreed-upon method for transmitting information between two devices. The *protocol* determines the communication method. An example of a *protocol* is TCP/IP.

**Client** - A process that makes requests to a Print Service. The *Client* process may reside in a Printer, Mobile Device, Web Portal or other device. The role of a *Client* is separate from the overall role of a device.

**TargetDevice** - The destination for the data created by a service. A printer is one example of a *Target Device*.

**Document** - A unique data stream that contains some content. This is typically something like a PDF file, a Word Document, a JPG image, or some other data type.

**URL** - Fully qualified Uniform Resource Locator. Conforms To RFC 2396.

**Authentication** - The process of identifying an individual, usually based on a username and password. In security systems, *authentication* is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. *Authentication* merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual.

**Authorization** - The process of granting or denying access to a network resource. Most computer security systems are based on a two-step process. The first stage is authentication, which ensures that a user is who he or she claims to be. The second stage is *authorization*, which allows the user access to various resources based on the user's identity.

**Object** - An object is a self-contained entity that contains data and methods to manipulate the data. Example objects are Printer, Job, and Document.

**Element** - An XML construct that defines an object, and the components of an object. In this Document *element* is also used to describe a characteristic of an object.

**Attribute** - An XML construct that defines an attribute of an element.

### 3 Print Service Interface Protocol

#### 3.1 PSI Protocol

A device or service that implements the Print Service Interface is required to implement the interface as WSDL 1.1 [WSDL1.1] defined SOAP 1.1 [SOAP1.1] method calls via HTTP 1.1 [HTTP1.1] or future backwards compatible versions of HTTP. Since the interface is defined by WSDL and Semantic Model schemas, the PSI Protocol is case sensitive. TLS [TLS] and SSL [SSL] support is optional for a Print Service, and a Target Device. A Print Service and a Target Device should try to fully support TLS, and if the client cannot negotiate a TLS session, downgrade to SSL. Service and Client should try to negotiate to the most secure connection possible. Future implementations of Services and Clients can utilize advances in the security specifications.

Without utilizing some form of authentication and authorization, a Print Service should not be deployed in a public space.

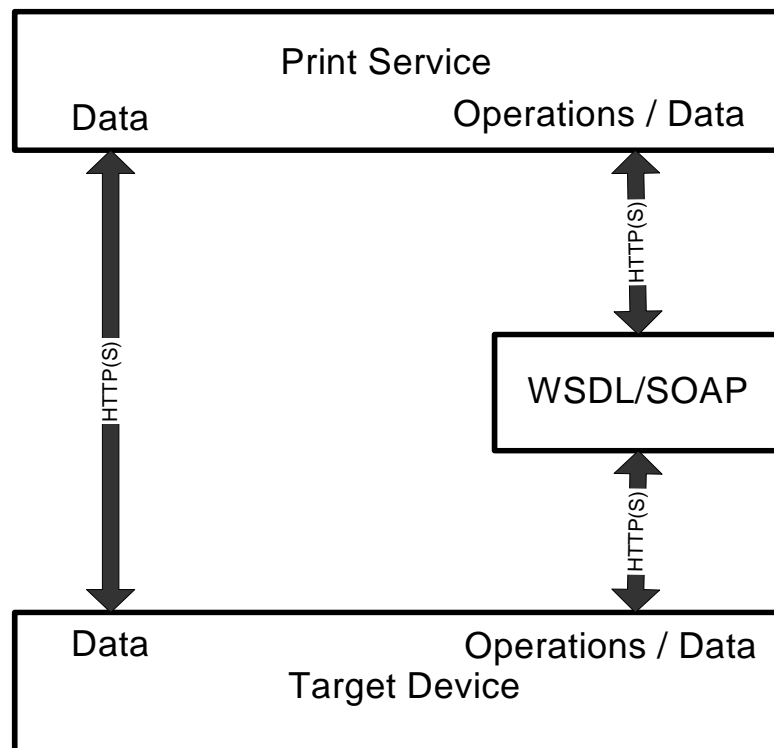


Figure 1 Protocol Connection Diagram

### 3.2 PSI Protocol Stack

7 – Application Layer:	PSI Client or PSI Server
6 – Presentation Layer:	WSDL/SOAP (mandatory)
5 – Session Layer:	HTTP (preferred) [TLS, SSL]
4 – Transport Layer	TCP (preferred)
3 – Network Layer	IP
2 – Data Link Layer	Ethernet, Bluetooth, etc.
1 – Physical Layer	Wire, Light, Radio, etc.

*Figure 2 PSI Protocol Stack*

## 4 Interface Definition

This section provides the Print Service Interface definition.

The normative abstract URI of the namespace for the PSI interfaces shall be (Note that this will change as the WSDL rolls until the PSI specification is a “PWG Standard”.)

namespace = <http://www.pwg.org/ps/psi1/>

The WSDL (WSDL 1.1) documents provide an example embodiment of this specification. For the current informative WSDL, see the following references. Note that these WSDL files are intended for development purposes only. Run time implementations shall not reference these definitions. WSDL definitions of this protocol specification shall utilize the version tag.

<http://www.pwg.org/schemas/ps/2005/02/JobControllInterface.wsdl>

<http://www.pwg.org/schemas/ps/2005/02/QueryEndPointsInterface.wsdl>

<http://www.pwg.org/schemas/ps/2005/02/ServiceCapabilitiesInterface.wsdl>

<http://www.pwg.org/schemas/ps/2005/02/TargetDeviceSupportInterface.wsdl>

Rationale: Current state of toolkits and WSDL editors do not allow for normative WSDL definitions, and thus we cannot publish normative references. The eventual goal is for normative WSDL to be published and referenced from this document before PSI becomes a full “PWG Standard”.



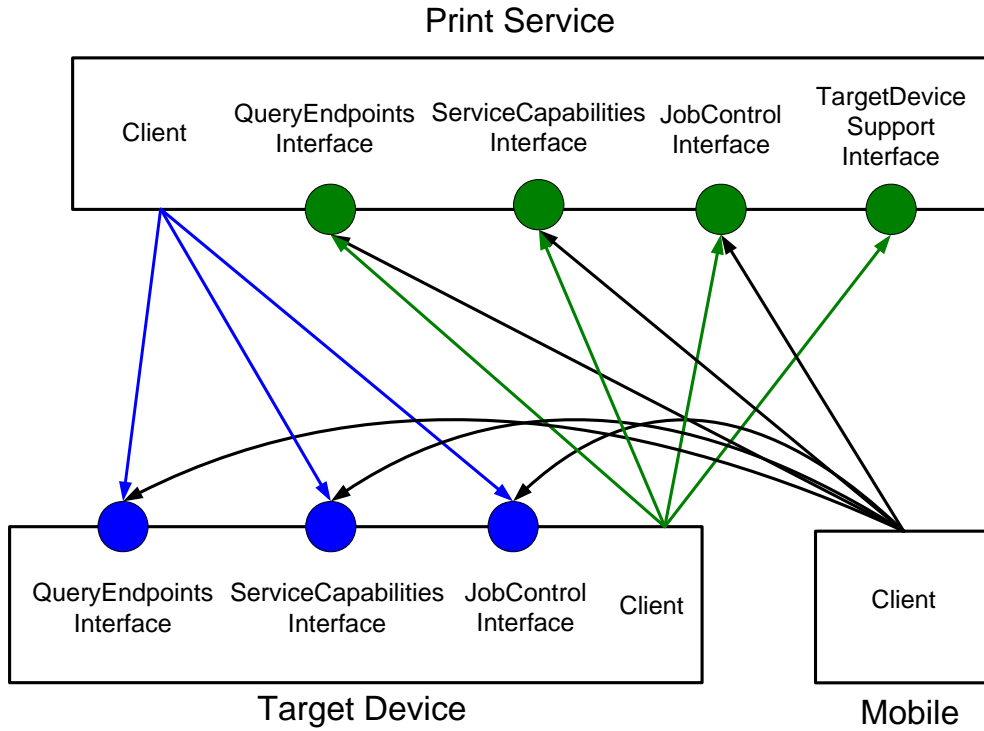


Figure 3 Interface Instantiation

Note: The circles represent interfaces.

## 4.1 PSI Service Root URL

Print Services and Target Devices must support discovery of various constructs via HTTP GET on the following well known paths:

http://<hostname>:3800/psi/<version>	returns URL of the QueryEndpointsInterface (required for a PS and TD to support)
http://<hostname>:3800/psi/<version>/wsdl	returns WSDL for the QueryEndpointsInterface (required for a PS and TD to support)
http://<hostname>:3800/psi/<version>/upnp-dev/1	returns UPnP 1.0 Device Description XML file (recommended for a PS and TD to support based upon deployment) (Later versions may add additional paths as necessary)
http://<hostname>:3800/psi/<version>/slp	returns SLP and LDAP advertised service attributes. (SLP Printer Template v2.0) (recommended for a PS and TD to support based upon deployment)
http://<hostname>:3800/psi/<version>/wsil	returns WSIL for the service (recommended for a PS and TD to support, moving to required in post 1.0)

*Figure 4 PSI Service Root URL Paths*

Note: <version> will always be of the form “<major>.<minor>”

- A Print Service shall implement PSI on port 3800 and may implement PSI on port 80.
- A Target Device shall implement a PSI Service only on port 3800. When a Target Device contacts a Print Service, it shall first try port 3800, and then if that fails, may utilize port 80.
- A Client shall always try to communicate with a Print Service on port 3800 first, and if that fails, then may utilize port 80.
- If the client knows the host, and the host does not respond to the Service Root URL, then the client may try a discovery mechanism to determine the Service Root URL for that specific host.
- PSI should be integrated with future discovery mechanisms. (ex. WS Addressing [WSAddressing])

## 4.2 Negotiating a secure PSI connection

A TLS connection can be upgraded within HTTP if the Print Service or Target Device supports TLS as specified in RFC 2817 [RFC2817].

See RFC 2817 [RFC2817] and RFC 2910 [RFC2910] section 8.1.2.

### 4.3 QueryEndPointsInterface

This interface allows a Client to discover the interfaces and versions of the interfaces that the service supports. The QueryEndPointsInterface itself is not included in the responses.

Interface Support:

- Print Service – Mandatory to Implement
- Target Device – Mandatory to Implement
- Client – Optional to utilize - service end-points can be administratively pre-configured

From this interface, Clients can find the additional interface end points that the Print Service exposes. It is required that all interfaces are exposed on port 3800.

Method Support:

- See relevant sub-section of Section 7 - Conformance

A Print Service can be queried for the following Interfaces:

1. name = ServiceCapabilitiesInterface
2. name = JobControlInterface
3. name = TargetDeviceSupportInterface

A Target Device can be queried for the following Interfaces:

1. name = ServiceCapabilitiesInterface
2. name = JobControlInterface

Vendors can define their own interface names and namespaces.

### 4.3.1 Interface Usage Examples

#### 4.3.1.1 *Bind to JobControlInterface*

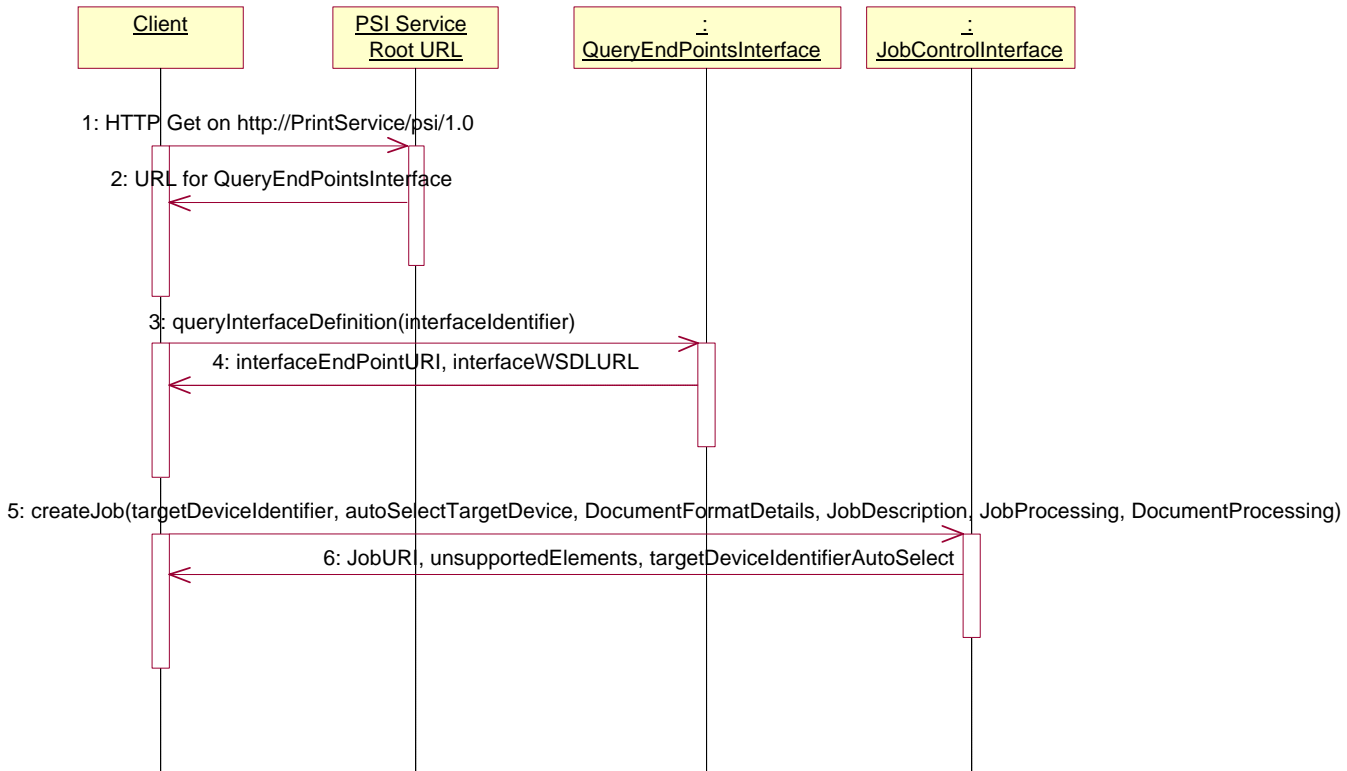


Figure 5 Bind to JobControlInterface

### 4.3.2 QuerySupportedInterfaces

QuerySupportedInterfaces () : interfaceIdentifiers : InterfacelIdentifierContainer

**Description:**

Allows a Client to determine the interfaces that a service supports.

**Returns:**

**interfaceIdentifiers : InterfacelIdentifierContainer** - A container of Interface Identifiers that describes the interfaces that the service supports. See Section [6.4].

**Exceptions:**

None

### 4.3.3 QueryInterfaceDefinition

QueryInterfaceDefinition (interfacelIdentifier : InterfacelIdentifier) : interfaceEndPointURI : URI,  
interfaceWSDLURL : URI

**Description:**

This method is used to retrieve the URI of an interface, and the http URL to the WSDL document for the requested interface.

**Returns:**

**interfaceEndPointURI : URI** – URI of the requested interface.

**interfaceWSDLURL : URI** – URL to the WSDL definition of the requested interface.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
interfacelIdentifier	Mandatory to Provide	Mandatory to Provide

interfacelIdentifier : InterfacelIdentifier

See Section 6.4.

Object Element Support:  
All elements are mandatory.

**Exceptions:**

Exception	Description
ClientErrorUnsupportedInterface	The Client provided an interfacelIdentifier that identifies an interface that the Print Service or Target Device does not support. An example of this is where the client requests a secure connection (<sec>true</sec>), and the Service / Device doesn't support a secure connection.

## 4.4 ServiceCapabilitiesInterface

The ServiceCapabilitiesInterface allows a Client to determine the capabilities of a Print Service or Target Device. The GetTargetDeviceElements method allows a Client to query for the Printer object elements that a particular Target Device supports. The GetKnownTargetDevices method allows a Client to discover the Target Devices that a service knows about. The ValidateReference method allows a Client to determine if a reference is valid, and accessible by the service.

Interface Support:

- Print Service – Mandatory to Implement
- Target Device – Mandatory to Implement
- Client – Optional to Use

Method Support:

- See relevant sub-section of Section 7 - Conformance

### 4.4.1 Interface Usage Examples

#### 4.4.1.1 *Client finds local color printers*

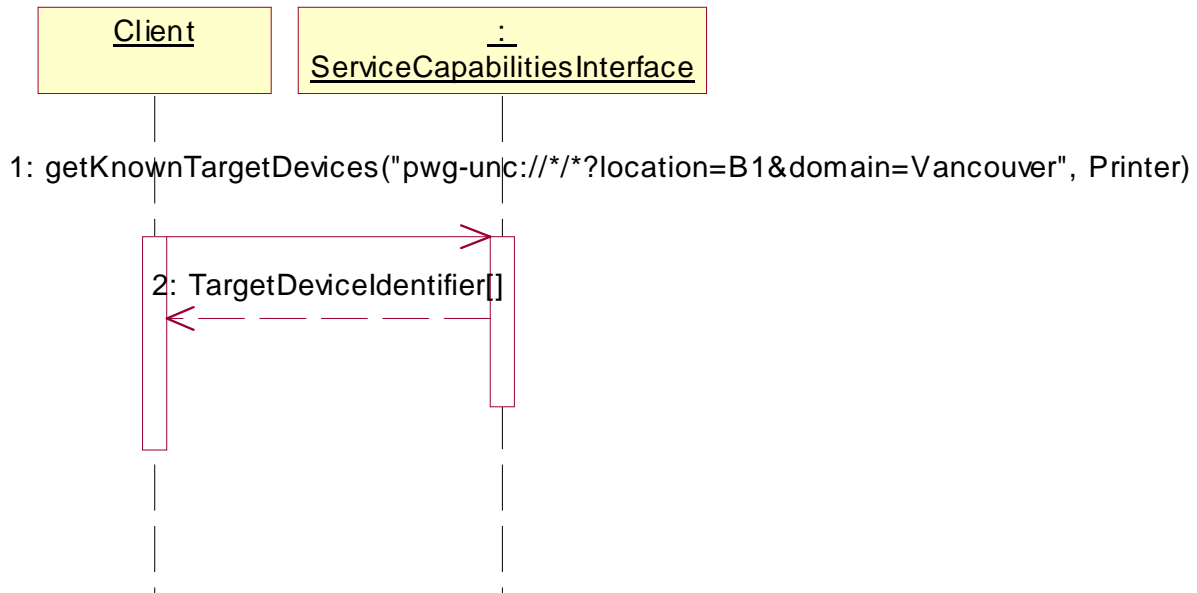


Figure 6 Client Finds Local Color Printers

4.4.1.2 Client determines a printer / service capabilities

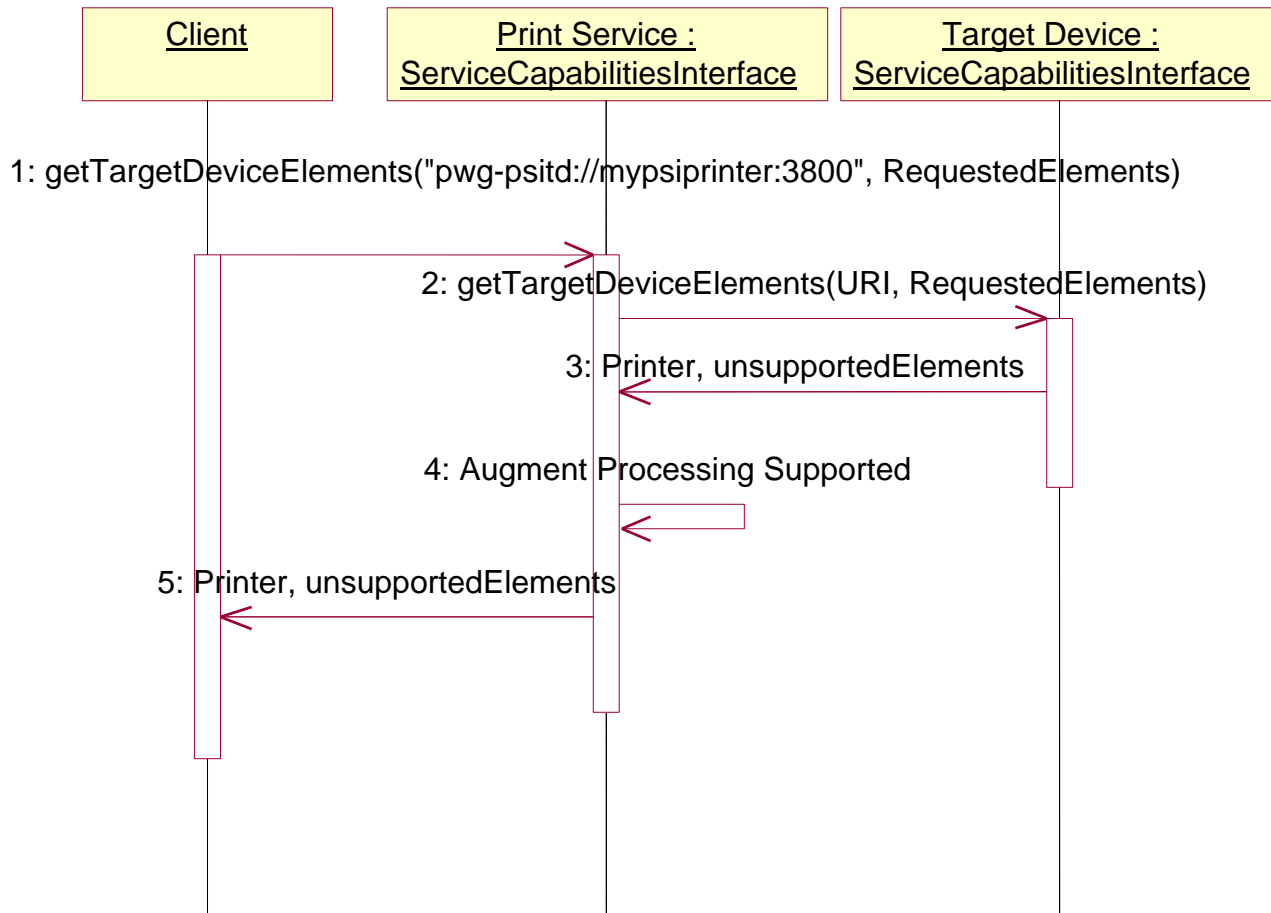


Figure 7 Client Determines a printer / service capabilities

**4.4.2 GetTargetDeviceElements**

GetTargetDeviceElements (targetDeviceIdentifier : URI, requestedPrinterElements : RequestedElements)  
 : printer : Printer, unsupportedElements : UnsupportedElements

**Description:**

Allows a Client to query the Print Service for the elements that a Target Device supports. It is required that the service return the aggregation of the elements that the Target Device and Print Service provides. Requested Elements that are not supported are not returned.

If this method is called on a Target Device, then the targetDeviceIdentifier parameter is validated by the Target Device

**Returns:**

**printer : Printer** - an object that conforms to the Semantic Model [PWG5105.1] schema Printer.xsd

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
targetDeviceIdentifier	Mandatory to Provide	Mandatory to Provide
requestedElements	Optional to Provide (can be null)	Optional to Provide (can be null)

**targetDeviceIdentifier : URI**

An URI that defines a Target Device. See definition in Section 6.2. If this method is called on a Target Device, then the targetDeviceIdentifier parameter shall be validated by the Target Device. If the Target Device type identified is not supported, then a ClientErrorUriSchemeNotSupported exception shall be thrown. If the Target Device is not reachable by the Print Service, or it has not registered itself with the Print Service then a ClientErrorNotFound exception shall be thrown.

**requestedPrinterElements : RequestedElements**

A TargetDevice object instance that contains the elements that the Client is querying for. If null, all supported elements shall be returned. If a client provides requestedElements, only the requestedElements that a service supports shall be returned. See the definition in the Semantic Model [PWG5105.1] Printer.xsd

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The Print Service can not reach the specified Target Device, or the specified Target Device has not registered itself with the Print Service.
ClientErrorUriSchemeNotSupported	A targetDeviceIdentifier type was provided that the Print Service does not support

**4.4.3 GetKnownTargetDevices**

GetKnownTargetDevices (targetDeviceIdentifierFilter : URI, printerFilterElements : Printer) :  
targetDeviceIdentifiers : TargetDeviceIdentifierContainer, unsupportedElements : UnsupportedElements



**Description:**

Queries the Print Service for the list of known Target Devices. If the method is called on a Target Device, then the Target Device returns a URIContainer with a single TargetDeviceIdentifier that represents itself. The Print Service may further restrict the returned targetDeviceIdentifiers based upon administrative configuration.

**Returns:**

**targetDeviceIdentifiers : TargetDeviceIdentifierContainer** – a container of TargetDeviceIdentifier URI's

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
targetDeviceIdentifierFilter	Optional to Provide (can be null)	Optional to Provide (can be null)
printerFilterElements	Optional to Provide (can be null)	Optional to Provide (can be null)

**targetDeviceIdentifierFilter : URI**

A TargetDeviceIdentifier URI that contains the information that must be matched when filtering the returned list of TargetDeviceIdentifiers. If this parameter is null, and the printerFilterElements is null, the service returns unfiltered Target Devices. See definition in Section 6.2.

The targetDeviceIdentifierFilter must follow a valid URI scheme, however wildcards (“\*”) are allowed in the server, port, and path parts, and the values of query parameters. Wildcards are not allowed in the scheme name. A wildcard is defined as an “\*” which means 0 to n characters, and otherwise matching is exact and case sensitive. The following are examples of a valid targetDeviceIdentifierFilter:

Filter	Description
pwg-psitd://*/psi/1.0?Location=*B1*	Filters on any pwg-psitd where Location=B1
fax:+360*	Filters on any Fax in the 360 area code
smb://printspooler.example.org/*	Filters on any smb share on printspooler.example.org

**printerFilterElements : Printer**

Any element value that is specified must be matched for the Target Devices whose current configuration is known to the Print Service . If this parameter is null, the service does not perform any filtering on the printerElements. (Give an example of a printerFilterElements)

Implementations are not intended to perform configuration queries on the Target Devices at the time of this method invocation.

Important printer filter elements could typically include:

- ProcessingSupported
  - resolution
- PrinterStatus
- ProcessingReady
  - mediaReady

See example in Section 15

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorUriSchemeNotSupported	The provided targetDeviceIdentifierFilter URI scheme is not supported.

**4.4.4 ValidateReference**

ValidateReference (reference : URI) : validReference : Boolean

**Description:**

Performs a reference and credential validation outside of the context of a Job. Allows a Client to validate that the Print Service has the ability to resolve the reference and if a user has provided the appropriate credentials to resolve the reference. Also, if the Print Service cannot support the document type referred to, a ClientErrorDocumentFormatNotSupported shall be thrown.

**Returns:**

**validReference : Boolean** – true if the reference is valid. An exception is thrown if the reference is not valid.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
reference	Mandatory to Provide	Mandatory to Provide

**reference : URI**

A URI that describes a reference to a source Document. See the definition in Section 6.3.

Object Element Support:  
Support of the URI “http:” reference element is Mandatory.

**Exceptions:**

<b>Exception</b>	<b>Description</b>
ClientErrorDocumentFormatNotSupported	The document format that the reference refers to is not supported.
ClientErrorNotAuthorized	The Print Service or Target Device is not authorized to retrieve the reference.
ClientErrorNotFound	The reference specified is not resolvable
ClientErrorUriSchemeNotSupported	The reference scheme provided is not supported.

## 4.5 JobControlInterface

This interface is implemented on both a Print Service and a Target Device. This is the mechanism to create a Job, add Documents to the Job, get status about the Job and the associated Documents, and to cancel the Job or the Documents within a Job.

It also provides the ability to specify how the Job and Documents are processed. A Client may retrieve Job and Document status as well as cancel individual Document processing, or the entire Job processing.

Document data can be provided either by pushing the Document data to the Print Service or Target Device, or by a reference to the Document. Pushed documents and by reference documents can co-exist within a Job. Documents can also consist of multiple files.

After performing an AddDocumentByPush operation, the client shall not perform any additional AddDocumentByReference, AddDocumentByValue or AddDocumentByPush operations until either the data has been delivered and a PushDocumentDataDelivered, or a CancelDocument call is made for the pending document. All other operations are allowed, except for CloseJob.

A secure connection from the Print Service to the Target Device can be specified by the client by providing a SecureConnectionRequired query parameter.

Interface Support:

- Print Service - Mandatory
- Target Device - Mandatory
- Client - Conditionally Mandatory - To create a Job, the Client must utilize this interface.

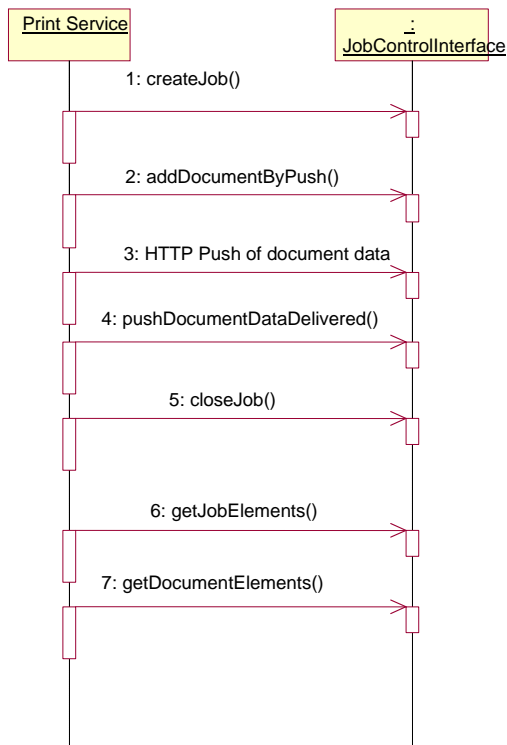
Method Support:

- See relevant sub-section of Section 7 - Conformance

### 4.5.1 Interface Usage Examples

#### 4.5.1.1 Print Service posts data to Target Device and queries status

This example shows the Print Service utilizing the PSI JobControlInterface on a Target Device to create a Job and deliver the data.



*Figure 8 Print Service posts data to Target Device*

**4.5.1.2 PSI Client initiates a Reference Job and queries status**

This shows a Client initiating a print by reference Job.

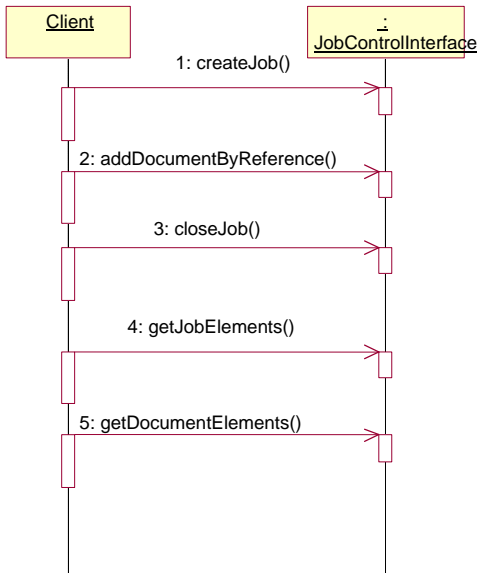


Figure 9 PSI Client initiates a Reference Job

**4.5.1.3 Print Service to Target Device by Reference**

This example shows how a Print Service can request that the Target Device pull the device ready data from the Print Service.

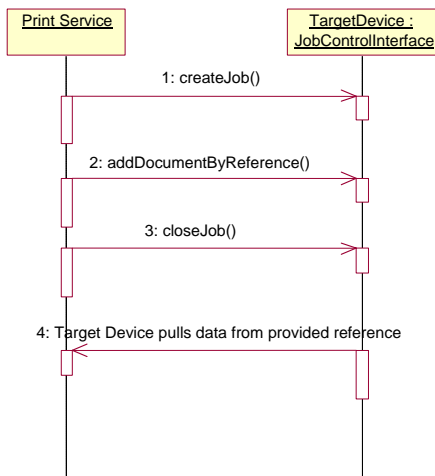


Figure 10 Print Service to Target Device by Reference

### 4.5.2 CreateJob

CreateJob (targetDeviceIdentifier : URI, autoSelectTargetDevice : Boolean, requestedTargetDeviceDataType : DocumentFormatDetails, jobDescription : JobDescription, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing) : jobURI : URI, unsupportedElements : UnsupportedElements, targetDeviceIdentifierAutoSelect : Boolean;

**Description:**

Create an instance of a Job.

This method shall be called before AddDocumentByPost, AddDocumentByValue or AddDocumentByReference is called. AddDocumentByPost, AddDocumentByValue and AddDocumentByReference calls can co-exist within a single Job.

If the Target Device is not reachable, the Print Service can choose to accept or reject the Job request. If it accepts the request, it can then wait for some amount of time for the Target Device to register with the Print Service and request the job with using FetchNextJob in the TargetDeviceSupportInterface. See behavior table in Section 5.

**Returns:**

**jobURI : URI** - Unique ID for the Job that must be utilized in subsequent calls. This value conforms to RFC 2396 [URI].

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**targetDeviceIdentifierAutoSelect : Boolean** – Indicates if the service will auto select or not.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
targetDeviceIdentifier	Optional to Provide (can be null)	Mandatory to Provide
autoSelectTargetDevice	Mandatory to Provide	Mandatory to Provide – shall be false
requestedTargetDeviceDataType	Optional to Provide (can be null)	Optional to Provide (can be null) - If provided, the Target Device validates.
jobDescription	Mandatory to Provide	Mandatory to Provide
jobProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)
defaultDocumentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)

**targetDeviceIdentifier : URI**

A URI that defines the Target Device that is to be associated with a Job. See definition in Section 6.2. If the method call is on a TargetDevice, the TargetDevice validates the parameter. See behavior table in Section 5.

If the client specifies the Print Service's Service Root URL as the targetDeviceIdentifier, then the Print Service is considered the final destination, and the Print Service can perform all job processing. For example, document transformation.

**autoSelectTargetDevice : Boolean**

If true, indicates that the Print Service should try and select a Target Device from the known Target Devices. See behavior table in Section 5.

**requestedTargetDeviceDataType : DocumentFormatDetails**

If this parameter is specified, then the Print Service shall transform the source Document into the data type requested. If the specified Target Device does not support the requested requestedTargetDeviceDataType, the Print Service shall throw a ClientErrorAttributesOrValuesNotSupported exception.

This may serve as a filter for autoSelect. For example if autoSelectTargetDevice is true and this parameter is non-null the Print Service should utilize this parameter in filtering the candidate Target Devices.

This object conforms to the DocumentFormatDetail object defined in the Semantic Model [PWG5105.1] PwgCommon.xsd. A repository of well known values for the elements of DocumentFormatDetails will eventually be published. Client and servers should utilize the well known values for interoperability.

Object Element Support:

Element	Support Description
documentFormat	Mandatory Mime Type
documentFormatVersion	Conditionally Mandatory if documentFormat is not sufficient.
documentFormatDeviceID	Conditionally Mandatory if documentFormat and documentFormatVersion is not sufficient.

**jobDescription : JobDescription**

An object that conforms to the Semantic Model [PWG5105.1] JobDescription.xsd.

Object Element Support:  
See Section 7.7.2

**jobProcessing : JobProcessing**

An object that conforms to the Semantic Model [PWG5105.1] JobProcessing.xsd.

Object Element Support:  
All elements are optional.



**defaultDocumentProcessing : DocumentProcessing**

An object defined by the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the default DocumentProcessing for each Document. Individual DocumentProcessing elements passed in to AddDocument\* methods override the values of these elements.

Object Element Support:  
All elements are optional.

**Exceptions:**

<b>Exception</b>	<b>Description</b>
ClientErrorAttributesOrValuesNotSupported	One of the requested processing parameters in jobProcessing or defaultDocumentProcessing is not supported by the Print Service/Target Device combination, and Fidelity was specified as true. This exception has an unsupportedElements object embedded within it.
ClientErrorDocumentFormatError	The requestedTargetDeviceDataType is not a format that is supported.
ClientErrorNotAuthorized	The client is not authorized to print on the provided targetDeviceIdentifier.
ClientErrorNotFound	The client provided a targetDeviceIdentifier that the Print Service can not reach.
ClientErrorUriSchemeNotSupported	The client provided a targetDeviceIdentifier that specifies a scheme that the Print Services does not support.
ServerErrorBusy	The Print Service/Target Device is too busy processing current jobs to accept a new job request.
ServerErrorDeviceError	The Target Device is in an error state.
ServerErrorNotAcceptingJobs	The Print Service/Target Device is not accepting Jobs at the current time.
ServerErrorServiceUnavailable	The Specified Target Device is not reachable, and the service does not perform queuing / retry.

**4.5.3 CloseJob**

CloseJob (jobURI : URI) : void

**Description:**

Indicates that no additional documents will be added to the job. Future AddDocumentByPost, AddDocumentByValue, or AddDocumentByReference calls will throw exceptions. This method will throw and ClientErrorDataPending exception if the data for a previous AddDocumentByPost has not been delivered. After CloseJob, once the Print Service knows the transformation data type, the Print Service may begin the transformations.

**Returns:**

void

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**Exceptions:**

Exception	Description
ClientErrorDataPending	The Job still has a document that is waiting for data.
ClientErrorInvalidUri	The jobURI provided is not a valid URI.
ClientErrorNotAuthorized	The client is not authorized to close the job.
ClientErrorNotFound	The jobURI provided does not have a corresponding Job.
ClientErrorNotPossible	The Job has been previously closed.

**4.5.4 AddDocumentByReference**

AddDocumentByReference (jobURI : URI, reference : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing) : documentNumber : int, unsupportedElements : UnsupportedElements

**Description:**

Adds a reference that identifies a Document to the Job. If the reference resolves to multiple files, the Document is then associated with multiple files. A reference can refer to multiple files - for example, an email can have multiple attachments associated with it.

**Returns:**

**documentNumber : int** – The document number in the Job. The documents begin numbering at 1. Conforms to definition in the Semantic Model [PWG5105.1].

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
Reference	Mandatory to Provide	Mandatory to Provide
documentDescription	Mandatory to Provide	Mandatory to Provide

documentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)
--------------------	--------------------------------------	--------------------------------------

**jobURI : URI**

JobURI - Unique ID for the Job. This value conforms to RFC 2396.

**reference : URI**

A URI that describes a reference to a source Document. See type definition in this document in Section 6.3. This parameter allows a Client to specify a reference to a Document.

Support of the URI http: reference is Mandatory on a Print Service and a Target Device.

**documentDescription : DocumentDescription**

The DocumentDescription to be used when creating a Document for the Job. An object that conforms to the Semantic Model [PWG5105.1] DocumentDescription.xsd.

Object Element Support:

See Section 7.8.2. The LastDocument element is ignored.

**documentProcessing : DocumentProcessing**

An object defined by the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the DocumentProcessing for each Document.

Object Element Support:

All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorAttributesOrValuesNotSupported	One of the requested processing parameters in documentProcessing is not supported by the Print Service / Target Device combination, and Fidelity was specified True. This exception has an unsupportedElements object embedded within it.
ClientErrorDataPending	Data from a previous AddDocumentByPush is still pending.
ClientErrorDocumentFormatNotSupported	The documentFormat specified in the documentDescription is not supported.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorJobNotAcceptingAdditionalDocuments	The Job has been closed.
ClientErrorNotAuthorized	The client is not authorized to add a Document to the specified Job.
ClientErrorNotFound	The jobURI specified is not found.

ClientErrorUriSchemeNotSupported	The reference scheme supplied is not supported.
ServerErrorMultipleDocumentJobsNotSupported	The Print Service or Target Device does not support multiple document jobs, and a Document has already been added to the Job.
ClientErrorDocumentAccessError	The service was unable to retrieve the referenced document due to authorization failure, or the document referred to is not found.

#### 4.5.5 AddDocumentByPush

AddDocumentByPush (jobURI : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing) : documentNumber : int, dataSinkURIs : DataSinkContainer, unsupportedElements : UnsupportedElements

**Description:**

This method allows a Client to add a Document to a Job. The Document data is subsequently delivered via a mechanism specified by the URL Scheme contained in one of the dataSinkURIs. A pushed Document can contain multiple files - for example, if a ZIP file, or a multi-part mime Document can be delivered to the service.

**Returns:**

**documentNumber : int** – The document number in the Job. The documents begin numbering at 1.

**dataSinkURIs : DataSinkContainer** – A container of dataSinkURIs by which a client can deliver the data to the server. Both a Print Service and a Target Device shall include a HTTP(S) URL that the Client can perform an HTTP Post of the Document data to, and this HTTP(S) URL shall always be the first element in the DataSinkContainer. This HTTP(S) URL conforms to RFC 2396. If the client has created a secure connection with the Print Service or Target Device, then it must be an HTTPS dataSinkURL on port 443.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentDescription	Mandatory to Provide	Mandatory to Provide
documentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentDescription : DocumentDescription**

The DocumentDescription to be used when creating an Document for the Job. An object that conforms to the Semantic Model [PWG5105.1] DocumentDescription.xsd.

Object Element Support:  
See Section 7.8.2. The LastDocument element is ignored.

**documentProcessing : DocumentProcessing**

An object that conforms to the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the DocumentProcessing for each Document.

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorAttributesOrValuesNotSupported	One of the requested processing parameters in documentProcessing is not supported by the Print Service/Target Device combination, and Fidelity was specified True. This exception has an unsupportedElements object embedded within it.
ClientErrorDataPending	Data from a previous AddDocumentByPush is still pending.
ClientErrorDocumentFormatNotSupported	The documentFormat specified in the documentDescription is not supported.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorJobNotAcceptingAdditionalDocuments	The Job has been closed.
ClientErrorNotAuthorized	The client is not authorized to add a Document to the specified Job.
ClientErrorNotFound	The jobURI specified is not found.
ServerErrorMultipleDocumentJobsNotSupported	The Print Service or Target Device does not support multiple document jobs, and a Document has already been added to the Job.

**4.5.6 PushDocumentDataDelivered**

PushDocumentDataDelivered (jobURI : URI, documentNumber : int, dataSinkURI : URI, documentOctetsSent : long) : void

**Description:**

Indicates that the client has successfully delivered the document data to the corresponding dataSinkURI. For example, an HTTP post shall have completed with a successful response before the client calls this method. If the data could not be successfully delivered, the client shall call the CancelDocument Method.

The Client shall only call this method after delivering the data to one of the dataSinkURIs provided in the AddDocumentByPush method response.

For dataSinkURI's where the client can-not determine when successful delivery has occurred (for example, SMTP), the documentOctetsSent parameter allows the client to inform the service of the size of the data delivered. This is useful for data sinks that may have a delayed delivery mechanism, or one that has a chunked delivery mechanism.

**Returns:**

void

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide
dataSinkURI	Mandatory to Provide	Mandatory to Provide
documentOctetsSent	Optional to Provide	Optional to Provide

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentNumber : int**

The document number in the Job.

**dataSinkURI : URI**

The URI that the client utilized to deliver the document data to. This value conforms to RFC 2396 [URI].

**documentOctetsSent : long**

The number of octets delivered. A value of 0 specifies an empty document.

**Exceptions:**

Exception	Description
ClientErrorInvalidUri	The jobURI specified is not valid.

ClientErrorNotAuthorized	The client is not authorized to add a Document to the specified Job.
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.

### 4.5.7 AddDocumentByValue

AddDocumentByValue (jobURI : URI, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, documentData : base64Binary) : documentNumber : int, unsupportedElements : UnsupportedElements

#### Description:

This method allows a Client to add a Document to a Job. The document data is delivered in-band in the method call in the documentData parameter. The Document can contain multiple files - for example, if a ZIP file, or a multi-part mime Document. This Document is a single binary blob base64Binary encoded.

#### Returns:

**documentNumber : int** – The document number in the Job.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

#### Parameters:

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentDescription	Mandatory to Provide	Mandatory to Provide
documentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)
documentData	Mandatory to Provide	Mandatory to Provide

#### **jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

#### **documentDescription : DocumentDescription**

The DocumentDescription to be used when creating an Document for the Job. An object that conforms to the Semantic Model [PWG5105.1] DocumentDescription.xsd.

Object Element Support:

See Section 7.8.2. The LastDocument element is ignored.

#### **documentProcessing : DocumentProcessing**

An object that conforms to the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the DocumentProcessing for each Document.

Object Element Support:  
All elements are optional.

**documentData : base64Binary**

Data for the document, base 64 encoded.

**Exceptions:**

Exception	Description
ClientErrorAttributesOrValuesNotSupported	One of the requested processing parameters in documentProcessing is not supported by the Print Service/Target Device combination, and Fidelity was specified True. This exception has an unsupportedElements object embedded within it.
ClientErrorDataPending	Data from a previous AddDocumentByPush is still pending.
ClientErrorDocumentFormatNotSupported	The documentFormat specified in the documentDescription is not supported.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorJobNotAcceptingAdditionalDocuments	The Job has been closed.
ClientErrorNotAuthorized	The client is not authorized to add a Document to the specified Job.
ClientErrorNotFound	The jobURI specified is not found.
ServerErrorMultipleDocumentJobsNotSupported	The Print Service or Target Device does not support multiple document jobs, and a Document has already been added to the Job.

**4.5.8 GetJobs**

GetJobs (jobFilterElements : Job, requestedJobElements : RequestedElements, limit : int) : jobs : JobContainer, callbackInterval : uint32, unsupportedElements : UnsupportedElements

**Description:**

Returns an array of Job objects that meet the filtering criteria specified in jobFilterElements. Each Job Instance returned, shall contain the elements specified in requestedElements that are supported by the Print Service. If a group level element is requested, then all of the elements within the group that the service supports must be returned. Only the jobs that the user is authorized for (either by implementation or by administrative policy) will be returned.

**Returns:**



**jobs : JobContainer** – A container of Jobs where each Job conforms to the Semantic Model [PWG5105.1] Job.xsd.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobFilterElements	Optional to Provide (can be null)	Optional to Provide (can be null)
requestedJobElements	Optional to Provide (can be null)	Optional to Provide (can be null)
limit	Mandatory to Provide	Mandatory to Provide

**jobFilterElements : Job**

Specifies the elements used as a filter when creating the Job list that is returned. This parameter is an instance of a Job object with any of the Job elements filled in for which a match is desired. For example Job.Description.JobOriginatingUserName, or Job.Status.JobID.

A service must match all elements which are specified in order for a job to be placed in the resultant job list. Also, only the elements which a service supports are used for matching.

Object Element Support:  
All elements are optional.

**requestedJobElements : RequestedElements**

The requested elements and element groups that may be included in each Job Instance. If no elements are specified, then all of the elements that the Print Service or Target Device supports are returned.

Object Element Support:  
All elements are optional.

**limit : int**

Limits the number of jobs returned to a specific number. 0 indicates no limit.

**Exceptions:**

Exception	Description
ClientErrorNotAuthorized	The client is not authorized to perform the operation.

### 4.5.9 GetJobElements

GetJobElements (jobURI : URI, requestedJobElements : RequestedElements) : job : Job, callbackInterval : uint32, unsupportedElements : UnsupportedElements

**Description:**

Queries the Print Service for a particular Job's Elements. The Job instance returned shall contain the elements specified in requestedJobElements that are supported by the Print Service. If a group level element is requested, then all of the elements within the group that the service supports (which includes the mandatory elements) shall be returned.

**Returns:**

**job : Job** - a Job instance that conforms to the Semantic Model [PWG5105.1] Job.xsd.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
requestedJobElements	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

JobURI - Unique ID for the Job. This value conforms to RFC 2396.

**requestedJobElements : RequestedElements**

Used to allow the Client to request which elements may be returned. See definition in the Semantic Model [SM] Job.xsd. If no elements are specified, then all of the elements that the Print Service, or Target Device supports are returned.

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job associated with it.
ClientErrorNotAuthorized	The client is not authorized to retrieve information about the requested Job.
ClientErrorInvalidUri	The jobURI specified is not valid.

**4.5.10 SetJobElements**

SetJobElements (jobURI : URI, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing) : unsupportedElements : UnsupportedElements

**Description:**

Allows a client to modify the JobProcessing, and DefaultDocumentProcessing elements within a Job. This method is only allowed before the job has begun processing.

**Returns:**

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
jobProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)
defaultDocumentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

JobURI - Unique ID for the Job. This value conforms to RFC 2396.

**jobProcessing : JobProcessing**

An object that conforms to the Semantic Model [PWG5105.1] JobProcessing.xsd.

Object Element Support:  
All elements are optional.

**defaultDocumentProcessing : DocumentProcessing**

An object defined by the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the default DocumentProcessing for each Document. Individual DocumentProcessing elements passed in to AddDocument\* methods override the values of these elements.

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job associated with it.
ClientErrorNotAuthorized	The client is not authorized to modify information about the requested Job.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorSequenceError	The job has already entered the processing state.

**4.5.11 CancelJob**

CancelJob (jobURI : URI, documentMessage : String) : void

**Description:**

Cancel the processing of a specific JobURI and all of the Documents associated with it. Throws an exception in an error condition.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentMessage	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentMessage : String**

Allows a Client to associate a human readable message with the Cancel request.  
Can be null or an empty string.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job associated

	with it.
ClientErrorNotAuthorized	The client is not authorized to retrieve information about the requested Job.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorNotPossible	It is not possible to cancel the job.
SuccessOkPartiallyCanceled	The Job was canceled, but some of the processing has already been completed and printed.

#### 4.5.12 GetDocuments

GetDocuments (jobURI : URI, documentFilterElements : Document, requestedDocumentElements : RequestedElements) : documents : DocumentContainer, callbackInterval: uint32, unsupportedElements : UnsupportedElements

##### Description:

Returns an array of Documents that are associated with the Job. Each Document Instance returned shall contain the elements specified in requestedElements that are supported by the Print Service. If a group level element is requested, then all of the elements within the group that are supported by the Print Service shall be returned.

##### Returns:

**documents : DocumentContainer** – Collection of documents. Each document conforms to the definition in the Semantic Model [PWG5105.1] Document.xsd.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

##### Parameters:

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentFilterElements	Optional to Provide (can be null)	Optional to Provide (can be null)
requestedDocumentElements	Optional to Provide (can be null)	Optional to Provide (can be null)

##### **jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentFilterElements : Document**

Specifies the elements used as a filter when creating the Document list that is returned. If null, all documents for the Job are returned. For example:

Document.DocumentDescription.DocumentName,  
Document.DocumentDescription.DocumentFormat.

A service must match all elements that are specified in order for a document to be placed in the resultant job list. Also, only the elements that a service supports are used for matching.

See example in Section 15.

**requestedDocumentElements : RequestedElements**

This parameter is the list of elements and element groups you want to come back in each Document.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job associated with it.
ClientErrorNotAuthorized	The client is not authorized to retrieve information about the requested Job.
ClientErrorInvalidUri	The jobURI specified is not valid.

**4.5.13 GetDocumentElements**

GetDocumentElements (jobURI : URI, documentNumber : int, requestedDocumentElements : RequestedElements) : document : Document, callbackInterval : uint32, unsupportedElements : UnsupportedElements

**Description:**

Queries the Print Service for a particular Document’s Elements. The Document instance returned will contain the elements specified in requestedDocumentElements. If a group level element is requested, then all of the elements within the group that the service supports (which includes the mandatory elements) must be returned.

**Returns:**

**document : Document** – a Document instance that conforms to the Semantic Model [PWG5105.1] Document.xsd.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide
requestedDocumentElements	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentNumber : int**

The document number in the Job.

**requestedDocumentElements : RequestedElements**

Used to allow the Client to specify which elements are to be returned.

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.
ClientErrorNotAuthorized	The client is not authorized to retrieve information about the requested Document.
ClientErrorInvalidUri	The jobURI specified is not valid.

**4.5.14 SetDocumentElements**

SetDocumentElements (jobURI : URI, documentNumber : int, documentProcessing : DocumentProcessing) : unsupportedElements : UnsupportedElements

**Description:**

Allows a client to modify documentProcessing elements. This method is only allowed before the job (not just the document) has begun processing.

**Returns:**

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide
documentProcessing	Optional to Provide (can be null)	Optional to Provide (can be null)

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentNumber : int**

The document number in the Job.

**documentProcessing : DocumentProcessing**

An object that conforms to the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the DocumentProcessing for each Document.

Object Element Support:  
All elements are optional.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.
ClientErrorNotAuthorized	The client is not authorized to modify information about the specified Document.
ClientErrorInvalidUri	The jobURI specified is not valid.
ClientErrorSequenceError	The job has already entered the processing state.

**4.5.15 CancelDocument**

CancelDocument (jobURI : URI, documentNumber : int, documentMessage : String) : void

**Description:**

Cancel the processing of a specific Document item.

**Parameters:**



Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide
documentMessage	Optional to Provide (can be null or an empty string)	Optional to Provide (can be null or an empty string)

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentNumber : int**

The document number in the Job.

**documentMessage : String**

Allows a Client to associate a human readable message with the Cancel request.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.
ClientErrorNotAuthorized	The client is not authorized to retrieve information about the requested Document.
ClientErrorInvalidUri	The jobURI specified is not valid.
SuccessOkPartiallyCanceled	The Job was canceled, but some of the processing has already been completed and printed.

**4.5.16 FetchDocumentDataByPull**

FetchDocumentDataByPull (jobURI : URI, documentNumber : int) : dataSourceURIs : DataSourceContainer

**Description:**

Allows a client to pull the transformed document data. This method will throw an exception in the situation where the transformation is not completed. The document data is subsequently retrieved via a mechanism specified in the dataSourceURIs. For Print Services that authenticates the users, this method must only be allowed for the originating user.

When the original source document contains multiple files, the output data stream is a single stream that contains all of the transformed data as a single stream.

**Returns:**

**dataSourceURIs : DataSourceContainer** – A container of dataSourceURIs by which a client can retrieve document data from the server. A Print Service shall include a HTTP URL that the Client can perform an HTTP Get of the Document data from in the first position of the container. This HTTP URL conforms to RFC 2396. If the client has created a secure connection with the Print Service, then an HTTPS dataSourceURI on port 443 shall also be included.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**documentNumber : int**

The document number in the Job.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job.
ClientErrorNotAuthorized	The client is not authorized to retrieve the data fro this document.
ClientErrorInvalidUri	The jobURI provided is not a valid URI
ClientErrorNotPossible	The Service has not completed the transformation of the particular document.

**4.5.17 PullDocumentDataFetched**

PullDocumentDataFetched (jobURI : URI, documentNumber : int, dataSourceURI : URI) : void

**Description:**

Indicates that the client has successfully retrieved the document data from the corresponding dataSourceURI. The Client shall call this method after retrieving the data. The data shall be available until this method has been called, or an administrative timeout is reached.

**Returns:**

void

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide
dataSourceURI	Mandatory to Provide	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**documentNumber : int**

The documentNumber in the containing Job.

**dataSourceURI : URI**

dataSourceURI – The URI that the client utilized to retrieve the document data from.

**Exceptions:**

Exception	Description
ClientErrorDataPending	Data from a previous FetchNextDocumentByPull is still pending.
ClientErrorInvalidUri	The jobURI or dataSourceURI is invalid.
ClientErrorNotAuthorized	The client is not authorized to call this method on this jobURI.
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.

**4.5.18 FetchDocumentDataByValue**

FetchDocumentDataByValue (jobURI : URI, documentNumber : int) : documentData : base64Binary

**Description:**

Allows a client to pull the transformed document data in band. This method will throw an exception in the situation where the transformation is not completed. The document data is subsequently retrieved via a mechanism specified in the dataSourceURIs. This method must only be allowed for the originating user, or an administrator.

When the original source document contains multiple files, the output data stream is a single stream that contains all of the transformed data as a single stream.

**Returns:**

**documentData : base64Binary** – Data for the document base 64 encoded.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
jobURI	Mandatory to Provide	Mandatory to Provide
documentNumber	Mandatory to Provide	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**documentNumber : int**

The document number in the Job.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job.
ClientErrorNotAuthorized	The client is not authorized to retrieve the data fro this document.
ClientErrorInvalidUri	The jobURI provided is not a valid URI

**4.5.19 FetchJobs**

FetchJobs (printService : URI, jobFilter : Job) : unsupportedElements : UnsupportedElements

**Description:**

Allows a client to request that a Target Device or a Print Service retrieve Jobs that meet the jobFilter criteria, and their associated documents from a Print Service. If this method is called on a Print Service, then the service is the final destination. The Target Device or Print Service shall try and establish a connection to the Print Service before returning, and throw a ClientErrorNotFound exception if the Print Service is not reachable. The client can then poll the GetJobs method to monitor the Target Device / Print Services progress on fetching jobs from the specified Print Service.

If required, the printService URI should be adorned with the appropriate query parameters as defined in Section 6.1.

**Returns:**

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service	Client to Target Device
printService	Mandatory to Provide	Mandatory to Provide
jobFilter	Mandatory to Provide	Mandatory to Provide

**printService : URI**

A URI that defines the Print Service to retrieve the Job from.

**jobFilter : Job**

An instance of a Job that is passed on to the Print Service, and then used by the Print Service to correctly identify the instances of Jobs requested. Important elements include JobURI, JobAccountingUserID, and JobPassword.

It is required that either JobURI, or JobAccountingUserID is supplied. JobPassword is required if the Job on the Print Service was created with a password by the Client when the Job was created. The JobPassword is used to authorize the Target Device to be associated with the Job.

Any other elements can be specified to refine the filtering applied.

**Exceptions:**

Exception	Description
ClientErrorInvalidUri	The printService specified is not valid.
ClientErrorNotAuthorized	The client is not authorized to cause the service to fetch a job.
ClientErrorNotFound	The printService specified is not found.

## 4.6 TargetDeviceSupportInterface

This interface allows for a Target Device to register itself with the Print Service. It also allows for association of a particular Target Device with a Job. This interface is only implemented by a Print Service. This interface also provides a mechanism for the Target Device to retrieve Job and Document data. The Target Device may also inform the Print Service of state changes. It is important to note that a single TargetDevice is associated with a Job and the Job's associated Documents.

This interface may be utilized for firewall traversal. It allows a Target Device that is within a firewall to connect to a Print Service that is outside of a firewall.

Interface Support:

- Print Service – Mandatory to Implement
- Target Device - Not Supported
- Target Device Client – Recommended to Use (necessary for firewall traversal)

Method Support:

- See relevant sub-section of Section 7 - Conformance

### 4.6.1 Interface Usage Examples

#### 4.6.1.1 Target Device Polls and Pulls Target Device data

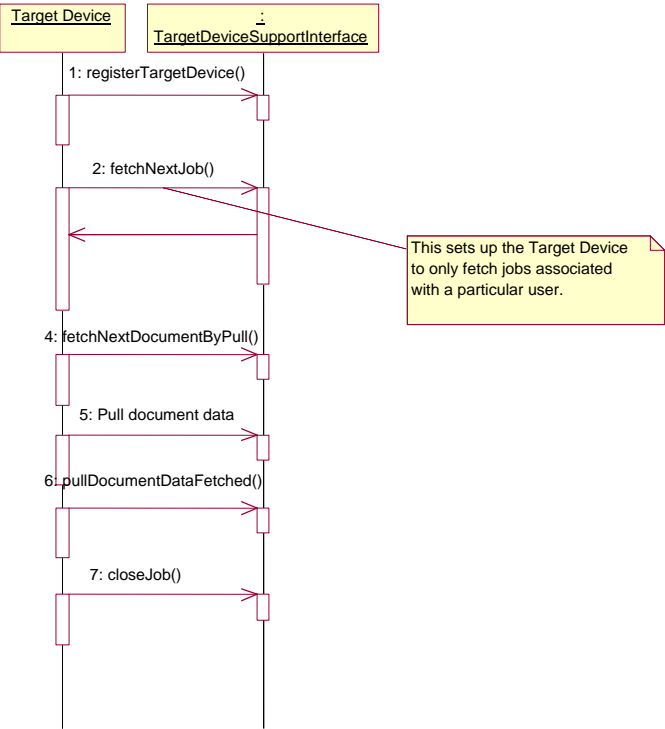


Figure 11 Target Device Polls

**4.6.1.2 Client Informs Target Device of Job**

In this scenario, the chances are good that there is a large delay between the Client creating a Job on the Print Service, and the Client informing the Target Device that it needs to fetch the Job. Also, the client has probably changed network location between the time it submitted the job to the time it requests the Target Device to Fetch the Job.

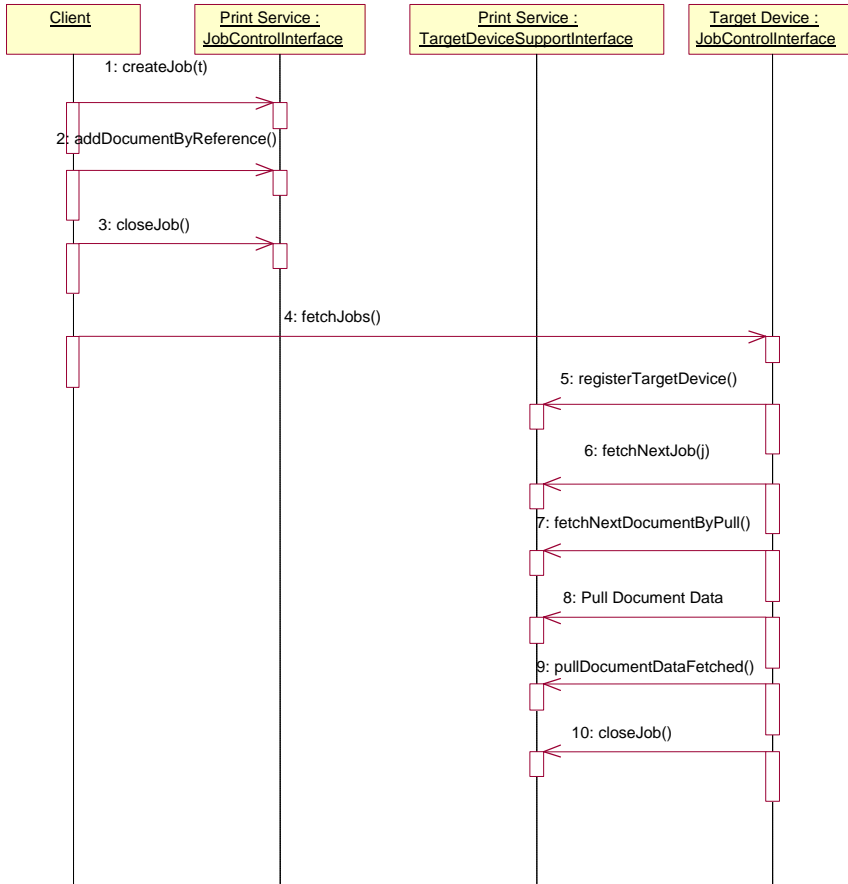


Figure 12 Client Informs Target Device of Job

**4.6.2 FetchNextJob**

FetchNextJob(jobFilter : Job, targetDeviceIdentifier : URI) : jobURI : URI, sendJobNotifications : JobEvents, jobDescription : JobDescription, jobProcessing : JobProcessing, defaultDocumentProcessing : DocumentProcessing, callbackInterval : uint32, unsupportedElements : UnsupportedElements

**Description:**



Allows a TargetDevice to determine if a Job is available for a specific Target Device. This method is intended to return the next available job that matches the job filter criteria. The RegisterTargetDevice method shall be called before this method is invoked. In a secure, behind the firewall deployment, a Print Service can choose to not require authentication credentials. In a public deployment, a Print Service should validate that the credentials provided here matches those provided during the RegisterTargetDevice call.

A Target Device shall first utilize the RegisterTargetDevice method before making any FetchNextJob method calls.

It is up to the Print Service to determine the appropriate ordering for the Job.

**Returns:**

**jobURI : URI** - Unique ID for the Job that must be utilized in subsequent calls. This value conforms to RFC 2396. [URI] If no Job is available, this method returns null for the jobURI parameter. This jobURI parameter shall be the same as the jobURI returned to the client that invoked the job via the JobControlInterface.

**sendJobNotifications : JobEvents** –A jobEvents element of type JobEvents as defined by the Semantic Model [PWG5105.1] Events.xsd. See Section 16

**jobDescription : JobDescription** - An object that conforms to the Semantic Model [PWG5105.1] JobDescription.xsd.

**jobProcessing : JobProcessing** - An object that conforms to the Semantic Model [PWG5105.1] JobProcessing.xsd.

**defaultDocumentProcessing : DocumentProcessing** - An object defined by the Semantic Model [PWG5105.1] DocumentProcessing.xsd. This parameter defines the default DocumentProcessing for each Document. Individual DocumentProcessing elements passed in to AddDocument\* methods override the values of these elements.

**callbackInterval : uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service
jobFilter	Mandatory to Provide
targetDeviceIdentifier	Mandatory to Provide

**jobFilter : Job**

An instance of a Job that is used by the Print Service to correctly identify the instances of Jobs that the Target Device is requesting. Important elements include JobURI, JobAccountingUserID, and JobPassword.

It is required that either JobURI, or JobAccountingUserID is supplied.

JobPassword is required if the Job on the Print Service was created with a password by the Client when the Job was created. The JobPassword is used to authorize the Target Device to be associated with the Job.

This parameter can be null.

**targetDeviceIdentifier : URI**

A URI that defines the Target Device that is to be associated with a Job for jobs that were not created with a particular TargetDeviceIdentifier, and the jobs for which the jobFilter matches; For Jobs that were created with a TargetDeviceIdentifier, this parameter also serves as a filter – only jobs that either did not have a TargetDeviceIdentifier, or that did have a TargetDeviceIdentifier, and it matches will be returned to this particular Target Device. See definition in Section 6.2.

**Exceptions:**

Exception	Description
ClientErrorNotAuthorized	The client is not authorized to close the specified Job.
ClientErrorInvalidUri	The targetDeviceIdentifier provided is not a valid URI.

**4.6.3 CloseJob**

CloseJob (jobURI : URI) : void

**Description:**

Indicates that the client has completed the job. Future FetchNextDocument calls will throw exceptions. CloseJob can be used by the Target Device to close the job even in the situation where the Target Device did not process all of the job’s associated documents.

**Returns:**

void

**Parameters:**

Parameter	Client to Print Service
jobURI	Mandatory to Provide

**jobURI : URI**

JobURI - Unique ID for the Job. This value conforms to RFC 2396 [URI].

**Exceptions:**

Exception	Description
ClientErrorDataPending	The Job still has a document that is waiting for data.
ClientErrorInvalidUri	The jobURI provided is not a valid URI.
ClientErrorNotAuthorized	The client is not authorized to close the specified Job.
ClientErrorNotFound	The jobURI provided does not have a corresponding Job.

**4.6.4 FetchNextDocumentByPull**

FetchNextDocumentByPull (jobURI : URI) : documentNumber : int, dataSourceURIs : DataSourceContainer, sendDocumentNotifications : DocumentEvents, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, lastDocument : Boolean, callbackInterval : uint32

**Description:**

Allows a TargetDevice to poll for Documents associated with a jobURI. This method is intended to return the next available document for the job. Returns 0 for the documentNumber if no document is available for the Job. The document data is subsequently retrieved via a mechanism specified in the dataSourceURIs.

**Returns:**

**documentNumber : int** - The documentNumber in the containing Job. If no Document is available, 0 is returned for documentNumber.

**dataSourceURIs : DataSourceContainer** – A container of dataSourceURIs by which a client can retrieve document data from the server. A Print Service shall include a HTTP URL that the Client can perform an HTTP Get of the Document data from in the first position of the container. This HTTP URL conforms to RFC 2396. If the client has created a secure connection with the Print Service, then an HTTPS dataSourceURI on port 443 shall also be included.

**sendDocumentNotifications : DocumentEvents** – A DocumentEvents element of type DocumentEvents as defined by the Semantic Model [PWG5105.1] Events.xsd. See Section 16.

**documentDescription : DocumentDescription** – An object that conforms to the definition in the Semantic Model [PWG5105.1] DocumentDescription.xsd

**documentProcessing : DocumentProcessing** - An object that conforms to the definition in the Semantic Model [PWG5105.1] DocumentProcessing.xsd.

**lastDocument : Boolean** – true if this is the last document for the job.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

**Parameters:**

Parameter	Client to Print Service
jobURI	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**Exceptions:**

Exception	Description
ClientErrorNotFound	The jobURI provided does not have a corresponding Job.
ClientErrorNotAuthorized	The client is not authorized to retrieve documents for the Job
ClientErrorInvalidUri	The jobURI provided is not a valid URI
ClientErrorDataPending	Data from a previous FetchNextDocumentByPull is still pending.

**4.6.5 PullDocumentDataFetched**

PullDocumentDataFetched (jobURI : URI, documentNumber : int, dataSourceURI : URI) : void

**Description:**

Indicates that the client has successfully retrieved the document data from the corresponding dataSourceURI. The Client shall call this method after retrieving the data.

**Returns:**

void

**Parameters:**

Parameter	Client to Print Service
jobURI	Mandatory to Provide
DocumentNumber	Mandatory to Provide
DataSourceURI	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**documentNumber : int**

The documentNumber in the containing Job.

**dataSourceURI : URI**

dataSourceURI – The URI that the client utilized to retrieve the document data from.

**Exceptions:**

Exception	Description
ClientErrorDataPending	Data from a previous FetchNextDocumentByPull is still pending.
ClientErrorInvalidUri	The jobURI or dataSourceURI is invalid.
ClientErrorNotAuthorized	The client is not authorized to call this method on this jobURI.
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentNumber does not exist in the job.

#### 4.6.6 FetchNextDocumentByValue

FetchNextDocumentByValue (jobURI : URI) : documentNumber : int, sendDocumentNotifications : DocumentEvents, documentDescription : DocumentDescription, documentProcessing : DocumentProcessing, lastDocument : Boolean, documentData : base64Binary, callbackInterval : uint32

##### Description:

Allows a TargetDevice to poll for Documents associated with a jobURI. This method is intended to return the next available document for the job. Returns 0 for the documentNumber if no document is available for the Job. The document data is returned in the documentData parameter encoded in base64Binary.

##### Returns:

**documentNumber : int** – The documentNumber in the containing Job. If no Document is available, 0 is returned for documentID.

**sendDocumentNotifications : DocumentEvents** – A DocumentEvents element of type DocumentEvents as defined by the Semantic Model [PWG5105.1] Events.xsd. See Section 16.

**documentDescription : DocumentDescription** - An object that conforms to the Semantic Model [PWG5105.1] DocumentDescription.xsd.

**documentProcessing : DocumentProcessing** - An object that conforms to the Semantic Model [PWG5105.1] DocumentProcessing.xsd

**lastDocument : Boolean** - True if this is the last document for the job.

**documentData : base64Binary** – Data for the document base 64 encoded.

**callbackInterval: uint32** – The number of seconds that a client should wait before calling the method again to check the status. Allowed values are 0 to max. 0 means that the client can call back whenever it wishes.

##### Parameters:

Parameter	Client to Print Service
jobURI	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist.
ClientErrorNotAuthorized	The client is not authorized to call this method on this jobURI.

**4.6.7 SendJobNotification**

SendJobNotification (jobURI : URI, targetDeviceIdentifier : URI, jobStatus : JobStatus, jobEvents : JobEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements

**Description:**

A Target Device shall call this method whenever a Job state change occurs - See the PWG common semantic model [PWG5105.1] for the state diagram. If the Target Device does not receive a valid response to the method invocation, it should retry the notification an administratively configured number of times, and it MUST utilize the same eventSequenceNumber so that the service can identify duplicate notifications.

**Returns:**

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service
jobURI	Mandatory to Provide
targetDeviceIdentifier	Mandatory to Provide
jobStatus	Mandatory to Provide
jobEvents	Optional to Provide
eventSequenceNumber	Mandatory to Provide

**jobURI : URI**

URI of the Job returned from FetchNextJob().

**targetDeviceIdentifier : URI**

Identifies the Target Device.

**jobStatus : JobStatus**

A JobStatus object that is defined by the Semantic Model [PWG5105.1] JobStatus.xsd

**jobEvents : JobEvents**

A jobEvents element of type JobEvents as defined by the Semantic Model [PWG5105.1] Events.xsd and in Section 16. The purpose is to send 1 or more events with each method invocation.

**eventSequenceNumber : int**

This is a numerically increasing number for the Job / Client pair as defined by the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist.
ClientErrorNotAuthorized	The client is not authorized to call this method on this jobURI.
ClientErrorInvalidUri	The jobURI or targetDeviceIdentifier is invalid.

**4.6.8 SendDocumentNotification**

SendDocumentNotification (jobURI : URI, documentNumber : int, targetDeviceIdentifier : URI, documentStatus : DocumentStatus, documentEvents : DocumentEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements

**Description:**

A Target Device shall call this method whenever a Document state change occurs - See the PWG common semantic model [PWG5105.1] for the state diagram. If the Target Device does not receive a valid response to the method invocation, it should retry the notification an administratively configured number of times, and it MUST utilize the same eventSequenceNumber so that the service can identify duplicate notifications.

**Returns:**

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service
jobURI	Mandatory to Provide
documentNumber	Mandatory to Provide
targetDeviceIdentifier	Mandatory to Provide
documentStatus	Mandatory to Provide
documentEvents	Optional to Provide
eventSequenceNumber	Mandatory to Provide

**jobURI : URI**

Unique ID for the Job returned from CreateJob. This value conforms to RFC 2396 [URI].

**documentNumber : int**

The document number in the Job.

**targetDeviceIdentifier : URI**

A URI that defines the Target Device. See definition in section 6.2.

**documentStatus : DocumentStatus**

A documentStatus object that conforms to the Semantic Model [PWG5105.1] DocumentStatus.xsd.

**documentEvents : DocumentEvents**

A documentEvents element of type DocumentEvents that conforms to the Semantic Model [PWG5105.1] Events.xsd and the description in Section 16. The purpose is to send 1 or more events with each method invocation.

**eventSequenceNumber : int**

This is a numerically increasing number for the Document / Client pair as defined by the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**Exceptions:**

Exception	Description
ClientErrorNotFound	The job specified by jobURI does not exist, or the documentID does not exist in the job.
ClientErrorInvalidUri	The jobURI or targetDeviceIdentifier is invalid.
ClientErrorNotAuthorized	The client is not authorized to call this method on this jobURI.



### 4.6.9 RegisterTargetDevice

RegisterTargetDevice (targetDeviceIdentifier : URI, targetDeviceElements : Printer, registrationLifetimeRequest : int, deviceWillPoll : Boolean) : registrationLifetimeGranted : int, notifyEvents : NotifyEvents, unsupportedElements : UnsupportedElements

**Description:**

Allows a Target Device to register itself with a Print Service. Throws an exception in an error condition. This method can be called by the Target Device more than once to renew the registration for another time period, or to communicate a change in capabilities. Also determines the mechanism (Active vs. Passive) for data to be transferred to the Target Device.

If the Target Device does not provide credentials, the Print Service can choose to allow or not allow the registration. In a secure environment, say behind a firewall, the Print Service can choose to allow un-authenticated registration. In a public environment, a Print Service should (strongly recommended) require authenticated registration.

**Returns:**

**registrationLifetimeGranted : int** – Returns actual registration time period in seconds.

**notifyEvents : NotifyEvents** – A notify event object defined by the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

**Parameters:**

Parameter	Client to Print Service
targetDeviceIdentifier	Mandatory to Provide
targetDeviceElements	Mandatory to Provide
registrationLifetimeRequest	Mandatory to Provide
deviceWillPoll	Mandatory to Provide

**targetDeviceIdentifier : URI**

A URI that defines the Target Device that is to be associated with a Job. See definition in Section 6.2.

**targetDeviceElements : Printer**

The Target Device constructs an instance of a Printer object, and fills in all of the appropriate elements. This allows the Target Device to communicate its capabilities to the service.

**registrationLifetimeRequest : int**

The Target Device will be registered for the amount of time specified by this parameter. It is up to the Target Device to re-register itself within the time period if it wants to remain registered. Duration in seconds forward from the time the method is invoked.

**deviceWillPoll : Boolean**

If true, it indicates that the Target Device shall request the transformed Document data from the PrintService. A Print Service shall wait for the Target Device to request the data for a minimum of 30 minutes.

If false, it indicates that the PrintService shall deliver the transformed Document data to the Target Device. A Print Service shall wait for the Target Device to request the data for a minimum of 30 minutes. This parameter is only utilized if the targetDeviceIdentifier is a pwg-psitd:// URI.

**Exceptions:**

Exception	Description
ClientErrorDocumentFormatNotSupported	The Print Service does not support transforming data into the format that the Target Device requires.
ClientErrorUriSchemeNotSupported	A targetDeviceIdentifier type was provided that the Print Service does not support
ClientErrorNotAuthorized	The client is not authorized to call this method.
ClientErrorInvalidUri	The targetDeviceIdentifier is invalid.

**4.6.10 UnregisterTargetDevice**

UnregisterTargetDevice (targetDeviceIdentifier : URI) : void

**Description:**

Unregisters the Target Device from the Print Service.

**Parameters:**

Parameter	Client to Print Service
targetDeviceIdentifier	Mandatory to provide

**targetDeviceIdentifier : URI**

A URI that defines the Target Device. See definition in Section 6.2.

**Exceptions:**

Exception	Description
ClientErrorUriSchemeNotSupported	A targetDeviceIdentifier type was provided that the Print

	Service does not support
ClientErrorNotAuthorized	The client is not authorized to call this method.
ClientErrorNotFound	The targetDeviceIdentifier was not previously registered.
ClientErrorInvalidUri	The targetDeviceIdentifier is invalid.

#### 4.6.11 SendTargetDeviceNotification

SendTargetDeviceNotification (targetDeviceIdentifier : URI, printerStatus : PrinterStatus, printerEvents : PrinterEvents, subunitEvents : SubunitEvents, vendorEvents : VendorEvents, eventSequenceNumber : int) : unsupportedElements : UnsupportedElements

##### Description:

Used by a TargetDevice to inform a Print Service of a Target Device state change. This method should be called whenever the Printer makes a state transition as defined in the PWG common Semantic Model [PWG5105.1]. If the Target Device does not receive a valid response to the method invocation, it should retry the notification an administratively configured number of times, and it MUST utilize the same eventSequenceNumber so that the service can identify duplicate notifications.

##### Returns:

**unsupportedElements : UnsupportedElements** – An UnsupportedElements object that contains the elements that are not supported by the service. Conforms to definition in the Semantic Model [PWG5105.1] PwgCommon.xsd. If an element itself was unsupported, the element is returned with an empty value. If the element was substituted to another value, the element with the original (unsupported) value is returned.

##### Parameters:

Parameter	Client to Print Service
targetDeviceIdentifier	Mandatory to provide
printerStatus	Mandatory to provide
printerEvents	Optional to provide
subunitEvents	Optional to provide
vendorEvents	Optional to provide
eventSequenceNumber	Mandatory to provide

##### **targetDeviceIdentifier : URI**

A URI that defines the Target Device. See definition in Section 6.2.

##### **printerStatus : PrinterStatus**

A printerStatus object of type PrinterStatus as defined in the Semantic Model [PWG5105.1] PrinterStatus.xsd.

##### **printerEvents : PrinterEvents**

A printerEvents element of type PrinterEvents as defined in the Semantic Model [PWG5105.1] Events.xsd and in Section 16. The purpose is to send 1 or more events with each method invocation.

**subunitEvents : SubunitEvents**

A subunitEvents element of type SubunitEvents as defined in the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**vendorEvents : VendorEvents**

A vendorEvents element of type VendorEvents as defined in the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**eventSequenceNumber : int**

This is a numerically increasing number for the Print Service / Client pair as defined by the Semantic Model [PWG5105.1] Events.xsd and in Section 16.

**Exceptions:**

Exception	Description
ClientErrorNotAuthorized	The client is not authorized to call this method.
ClientErrorNotFound	The targetDeviceIdentifier was not previously registered.
ClientErrorInvalidUri	The targetDeviceIdentifier is invalid.

## 5 Print Job Flows enabled by PSI

There are various ways that the Target Device data can be delivered to the Target Device, and that the Target Device for the Job can be selected. The Print Service can either push the Target Device data to the Target Device, or the Target Device can pull the data from the Print Service. The Target Device can be specified by the client, or the Target Device can be automatically selected by the Print Service. The behaviour of this selection can be configured by the Administrator. The following tables describe the various parameters to CreateJob and the RegisterTargetDevice methods, the Administrative configuration, and the resultant Print Service behavior.

autoSelectTargetDeviceSupported – (enum: false, true) into SM as a service attribute / job attribute

targetDeviceIdentifierSupported -

In the following tables, “Missing” means that the parameter provided was either null or an empty string.

### 5.1 Print Service is configured for target device specification and does not auto select:

autoSelectTargetDeviceSupported: False

targetDeviceIdentifierSupported: TrueFalse

CreateJob Parameters		Register Target Device Parameters	Result
Target Device Identifier	autoSelect Target Device	deviceWillPoll	
Legacy	Ignored	N/A	Print Service will push to Legacy Target Device. If Print Service can not reach the Target Device, the service has the choice to queue the job and wait until it can reach the device, or to cancel the job after a timeout, or fail the job immediately and throw a ServerErrorServiceUnavailable exception.
pwg-psitd	Ignored	Not Registered Yet	Print Service will try and deliver to the specified pwg-psitd. If the pwg-psitd is not reachable, Print Service can wait for registration, or periodically retry, or cancel the job after a timeout, or fail the job immediately and throw a ServerErrorServiceUnavailable exception.
pwg-psitd	Ignored	false	Print Service will try to push to pwg-psitd. If it can't, the job can periodically retry, or cancel the job after a timeout, or fail the job immediately and throw a ServerErrorServiceUnavailable exception.
pwg-psitd	Ignored	true	Print Service will wait for pwg-psitd to poll via the TargetDeviceSupportInterface.
Missing	false	false	Since the Target Device does not poll via FetchJobs, no association is ever made between the Job and the Target

			Device. If the client wants the data to be Pushed, they need to specify the targetDeviceIdentifier. In this situation, the Print Service will eventually time out the Job.
Missing	false	true	Print Service will wait for registration, and where the FetchJobs jobFilter matches, and then let pwg-psitd poll via the TargetDeviceSupportInterface.
Missing	true	false	A ClientErrorForbidden is thrown
Missing	true	true	A ClientErrorForbidden is thrown
Wild card specification	Ignored	N/A	A ClientErrorForbidden is thrown

### 5.2 Print Service is not configured for target device specification and always auto selects:

autoSelectTargetDeviceSupported: true

targetDeviceIdentifierSupported: false

CreateJob Parameters		Register Target Device Parameters	Result
TargetDevice Identifier	autoSelect Target Device	deviceWillPoll	
Legacy	Ignored	N/A	A ClientErrorForbidden is thrown
pwg-psitd	Ignored	Not Registered Yet	A ClientErrorForbidden is thrown
pwg-psitd	Ignored	false	A ClientErrorForbidden is thrown
pwg-psitd	Ignored	true	A ClientErrorForbidden is thrown
Missing	false	false	A ClientErrorForbidden is thrown
Missing	false	true	A ClientErrorForbidden is thrown
Missing	true	false	Print Service will select a Target Device from the known Target Devices that can fulfil the Job and Document Processing options. The data will be pushed to Target Device.
Missing	true	true	Print Service will select a Target Device from the known Target Devices that can fulfil the Job and Document Processing options. The Print Service will wait for the Target Device to pull the Job.
Wild card specification	Ignored	N/A	A ClientErrorForbidden is thrown.

### 5.3 Print Service is configured for target device specification and auto selection:

autoSelectTargetDeviceSupported: TrueFalse

targetDeviceIdentifierSupported: TrueFalse

CreateJob Parameters		Register Target Device Parameters	Result
Target Device Identifier	autoSelect Target Device	deviceWillPoll	
Legacy	Ignored	N/A	Print Service will push to Legacy Target Device.
pwg-psitd	Ignored	Not Registered Yet	Print Service will try and deliver to the specified pwg-psitd. If the pwg-psitd is not reachable, Print Service can wait for registration.
pwg-psitd	Ignored	false	Print Service will try to push to pwg-psitd. If it can't, the job needs to fail.
pwg-psitd	Ignored	true	Print Service will wait for pwg-psitd to poll via the TargetDeviceSupportInterface.
Missing	false	false	Since the Target Device does not poll via FetchJobs, no association is ever made between the Job and the Target Device. If the client wants the data to be Pushed, they need to specify the targetDeviceIdentifier. In this situation, the Print Service will eventually time out the Job.
Missing	false	true	Print Service will wait for registration, and where the FetchJobs jobFilter matches, and then let pwg-psitd poll via the TargetDeviceSupportInterface.
Missing	true	false	Print Service will select a Target Device from the known Target Devices that can fulfil the Job and Document Processing options. The data will be pushed to Target Device.
Missing	true	true	Print Service will select a Target Device from the known Target Devices that can fulfil the Job and Document Processing options. If selection is a pwg-psitd, then the Print Service will wait for the Target Device to pull the Job.
Wild card specification	Ignored	N/A	Print Service will select a Target Device from the known Target Devices that meet the filter requirements, and can fulfil the Job and Document Processing options.

## 6 Type Definitions

This section serves to provide examples of the parameters used in the interfaces.

Note: Whenever a password is specified in a Target Device Identifier URL, or a Reference URL, it should be done securely – either behind a firewall, or encrypted on a secure channel.

### 6.1 PWG Standard URI Query Parameters

The following PWG standard URI query parameters are defined for use with any URL scheme in specifying Target Device Identifiers (Section 7.2), Reference URI (Section 7.3), or any other URI.

Conformance:

- (1) A PSI Client MUST specify each parameter keyword in lowercase.
- (2) A PSI Client MUST specify well-known parameter values in lowercase.
- (3) A PSI Client MAY specify string parameter values in mixed case.

#### 6.1.1 'attid'

'attid' = Attachment id(s) comma separated, e.g.,

imap://joe;auth=login@foo.com:143?sec=ssl3&pw=bingo&msgid=123&attid=1.doc,2.doc,etc

#### 6.1.2 'auth'

Client authentication mechanism required to access this URI, e.g.,

ipp://bar.com/printer?auth=digest

Well-known values defined in section 4.4.2 of [RFC2911] are:

'none': no user authentication (i.e., 'anonymous').

'login': user specified in login protocol operation. (Same as 'requesting-user-name' in [IPP])

'basic': HTTP basic authentication [RFC2617].

'digest': HTTP digest authentication [RFC2617].

'certificate': user authenticated via certificate.

#### 6.1.3 'binary'

Binary transfer mode required to access this URI, e.g.,



```
ftp://joe:blues@myhost.com:21/somefile.pdf?passive=true&binary=true
```

Well-known values are:

'true': Use binary transfer mode.

'false': Use text transfer mode (default, when absent).

#### **6.1.4 'cert'**

Client certificate URI required to access this URI, e.g.,

```
ipp://bar.com/printer?auth=certificate&sec=tls&cert=ldap://f.com/abc
```

#### **6.1.5 'domain'**

Domain name for any non-Internet URI scheme, e.g.,

```
pwg-unc://*/?*?domain=Accounting
```

#### **6.1.6 'legacy'**

Legacy Target Device Identifier specified as a URI for PSI proxy, e.g.,

```
pwg-psitd://ps.foo.com/psi/1.0?legacy=pwg-unc://server/share
```

The 'legacy' query parameter is utilized by a Print Service to identify a particular instance of a Legacy Target Device. This can be used to expose existing Printers, Spoolers, Fax machines, etc. to PSI aware clients that expect to be able to talk directly to a Target Device.

#### **6.1.7 'location'**

Human-readable geographic location referenced by this URI, e.g.,

```
http://pwg-psitd://foo.com/psi/1.0?location=First%20Floor
```

#### **6.1.8 'msgid'**

The message id that identifies the particular message to be accessed by the uri, e.g.,

```
pop://joe;auth=login@foo.com:110?sec=ssl3&pw=bingo&msgid=123&attid=1.doc,2.doc,etc
```

#### **6.1.9 'name'**

Human-readable friendly name referenced by this URI, e.g.,

```
http://pwg-psitd://foo.com/psi/1.0?name=Dave's%20Printer
```

#### **6.1.10 'passive'**

Passive mode required to access this URI, e.g.,

```
ftp://joe:blues@myhost.com:21/somefile.txt?passive=true
```

Well-known values are:

'true': Use passive mode.

'false': Do not use passive mode (default, when absent).

### 6.1.11 'pw'

Password required to access this URI, e.g.,

pop://joe;auth=login@foo.com:110?sec=ssl3&pw=bingo&msgid=123

### 6.1.12 'sec'

Security mechanism required to access this URI, e.g.,

ipp://bar.com/printer?auth=digest&sec=tls

Well-known values defined in section 4.4.3 of [RFC2911] are:

'none': no security mechanism (i.e., 'anonymous').

'ssl3': SSL3 [SSL] security for communications channel.

'tls': TLS [RFC2246] security for communications channel.

### 6.1.13 'user'

Username required to access this URI, e.g.,

pop://joe;auth=login@foo.com:110?sec=ssl3&pw=bingo&msgid=123

## 6.2 Target Device Identifier

The Target Device Identifier allows for Clients to specify legacy as well as PSI Target Devices as Target Devices. It is up to the Print Service to provide the proxy support for legacy devices if it so chooses.

The Target Device Identifier is specified by a URI. The client is allowed to specify any existing, or yet to be defined URI schemes that refer to a potential Target Device. Below is a list of existing URI schemes that should be taken into consideration when implementing a Print Service. (Note: Novell PServer and AppleTalk PAP URI Schemes are intended to be defined in a future PWG Standards track document.)

Any of the references listed in the Reference type definition section are also allowed to be specified as a Target Device Identifier.

For other legacy URI schemes, see Appendix A – Legacy URI schemes (Informative).

Example Target Device Identifiers follow:

### 6.2.1 raw-tcp

Example:

```
raw-tcp://ourgroupprinter:9100
```

Notes:

REQUIRED 'host' or 'ip-addr' MUST be specified

REQUIRED 'port' MUST be specified (cannot be defaulted)

Background:

See the "raw-tcp:" URL scheme defined in the IANA-registered SLPv2 "direct print" template in the directory: <ftp://ftp.iana.org/assignments/svrlloc-templates>, in the file `printer_raw-tcp.1.0.en` which extends the base SLPv2 Printer Template v2.0 `service:printer:raw-tcp//<host>.<port>`

### 6.2.2 fax

Example:

```
fax:+358.555.1234567
```

Background:

See "fax:" URL scheme defined in "URLs for Telephone Calls" [RFC 2806].

### 6.2.3 pwg-psips, pwg-psitd

A URL to the Service Root URL of a Print Service, or a Target Device.

Examples:

```
pwg-psips://myspsitarget.mydomain.com:3800/psi/<version>
```

```
pwg-psitd://myspsitarget.mydomain.com:3800/psi/<version>
```

Notes:

REQUIRED 'host' or 'ip-addr' MUST be specified

pwg-psips: REQUIRED 'port' MUST be PWG PSI IANA-assigned value of 3800, or port 80.

pwg-psitd: REQUIRED 'port' MUST be PWG PSI IANA-assigned value of 3800 only.

If a Print Service wishes to expose a proxy PSI Target Device for a legacy Target Device, it may do so by adding the 'legacy' query parameter.

Example:

```
pwg-psitd://printservice.mydomain.com:3800/psi/<version>? legacy =smb://server/share
```

### 6.2.4 ipp

A URL to an IPP service.

Example:

```
ipp://bar.com:631/printer?auth=digest&sec=tls
```

Notes:

- (1) REQUIRED 'host' here is 'bar.com'
- (2) OPTIONAL 'port' here is '631' (default for IPP)
- (3) OPTIONAL 'path' here is 'printer'
- (4) PSI standard query 'auth' here is 'digest' [RFC2617]
- (5) PSI standard query 'sec' here is 'tls' [RFC2246]

### 6.2.5 ippfax

A URL to an IPPFAX service.

Example:

```
ippfax://foo.com/faxprinter?auth=certificate
```

Notes:

- (1) REQUIRED 'host' here is 'foo.com'
- (2) OPTIONAL 'path' here is 'faxprinter'
- (3) PSI standard query 'auth' here is 'certificate'

### 6.2.6 lpr

A URL to a lpr service.

Example:

```
lpr://bar.com:515/printer?location=First%20Floor
```

Notes:

- (1) REQUIRED 'host' here is 'bar.com'
- (2) OPTIONAL 'port' here is '515' (default for LPR)
- (3) OPTIONAL 'path' here is 'printer'
- (4) PSI standard query 'location' here is 'First Floor'

## 6.3 Reference

A reference is a URL that can refer to a single document, or multiple documents.

For other legacy URI schemes, see Appendix A – Legacy URI schemes (Informative).

Example references follow:

### 6.3.1 http

Example:

`http://www.pwg.org`

### 6.3.2 ftp

Examples:

`ftp://myhost.com:21/somefile.txt?passive=true&binary=false`

`ftp://myhost.com:21/somefile.pdf?passive=true&binary=true`

`ftp://ira:blues@myhost.com:21/somefile.pdf?passive=true&binary=true`

Notes:

- (1) OPTIONAL 'user' here is 'joe'
- (2) OPTIONAL 'password' here is 'blues'
- (3) REQUIRED 'host' here is 'myhost.com'
- (4) OPTIONAL 'port' here is '21' (the default for FTP)
- (5) OPTIONAL 'fpath' here is 'somefile.txt'

Background:

From section 5 of "Uniform Resource Locators (URL)" (RFC 1738):

`Ftpurl = "ftp://" login [ "/" fpath [ ";type=" ftptype ] ]`

`Fpath = fsegment *[ "/" fsegment ]`

`Fsegment = *[ uchar | "?" | "." | "@" | "&" | "=" ]`

`Ftptype = "A" | "I" | "D" | "a" | "i" | "d"`

`Login = [ user [ ":" password ] "@" ] hostport`

`Hostport = host [ ":" port ]`

### 6.3.3 file

Example:

file://myhost.com/myreference.pdf

Notes:

(1) OPTIONAL 'host' can be empty or "localhost" (but this is not a safe reference to pass over the network)

(2) REQUIRED 'path' MUST specify a specific file

Background:

From section 3.10 of "Uniform Resource Locators (URL)" (RFC 1738):

The file URL scheme is used to designate files accessible on a particular host computer. This scheme, unlike most other URL schemes, does not designate a resource that is universally accessible over the Internet.

A file URL takes the form:

file://<host>/<path>

where <host> is the fully qualified domain name of the system on which the <path> is accessible, and <path> is a hierarchical directory path of the form <directory>/<directory>/.../<name>.

### 6.3.4 pop

Example:

pop://joe;auth=<sasl-info>@mailhost.com:110

(1) (2) (3) (4)

pop://joe;auth=login@foo.com:110?sec=ssl3&pw=bingo&msgid=123&attid=1.doc,2.doc,etc

pop://joe;auth=login@host.com:110?sec=none&pw=password&msgid=123&attid=1.doc,2.doc,etc

(1) (2) (3) (4) (5) (6) (7) (8)

1 - username

2 - login

3 - host

4 - port

5 - security

- 6 - password
- 7 - message id
- 8 - attachment id(s)

Notes:

- (1) OPTIONAL 'user' parameter MUST be specified for PSI usage
- (2) OPTIONAL 'auth' parameter MAY be specified for APOP or SASL info
- (3) REQUIRED 'host' parameter SHOULD NOT be <ip-addr> for PSI usage
- (4) OPTIONAL 'port' parameter defaults to 110

Background:

From section 3 of "POP URL Scheme", RFC 2384, August 1998: "The POP URL scheme designates a POP server, and optionally a port number, authentication mechanism, authentication ID, and/or authorization ID.

The POP URL follows the common Internet scheme syntax as defined in RFC 1738 [BASIC-URL] except that clear text passwords are not permitted. If :<port> is omitted, the port defaults to 110. The POP URL is described using [ABNF] in Section 8.

A POP URL is of the general form:

pop://<user>;auth=<auth>@<host>:<port>

Where <user>, <host>, and <port> are as defined in RFC 1738, and some or all of the elements, except "pop://" and <host>, may be omitted."

### 6.3.5 imap

Example:

imap://joe;auth=login@foo.com:143?sec=ssl3&pw=bingo&msgid=123&attid=1.doc,2.doc,etc

imap://username;auth=login@server:port/mailbox;msgid=123?sec=ssl  
&pw=password&attid=1.doc,2.doc,etc

- (1)
- (2)
- (3)
- (4)
- (5)
- (6)
- (7)
- (8)
- (9)

- 1 - username
- 2 - ignored for psi
- 3 - host
- 4 - port

- 5 - mailbox
- 6 - message id
- 7 - use SSL to connect to mail server
- 8 - password
- 9 - attachment id(s)

From section 10 of "IMAP URL Scheme", RFC 2192, September 1997:

The IMAP URL:

```
<imap://minbari.org/gray-council;UIDVALIDITY=385759045;/UID=20>
```

Results in the following client commands:

```
<connect to minbari.org, port 143>
```

```
C: A001 LOGIN ANONYMOUS sheridan@babylon5.org
```

```
C: A002 SELECT gray-council
```

```
<client verifies the UIDVALIDITY matches>
```

```
C: A003 UID FETCH 20 BODY.PEEK[]
```

Background:

From section 2 of "IMAP URL Scheme", RFC 2192, September 1997:

"The IMAP URL scheme is used to designate IMAP servers, mailboxes, messages, MIME bodies [MIME], and search programs on Internet hosts accessible using the IMAP protocol.

The IMAP URL follows the common Internet scheme syntax as defined in RFC 1738 [BASIC-URL] except that clear text passwords are not permitted. If :<port> is omitted, the port defaults to 143.

An IMAP URL takes one of the following forms:

```
imap://<iserver>/
```

```
imap://<iserver>/<enc_list_mailbox>;TYPE=<list_type>
```

```
imap://<iserver>/<enc_mailbox>[uidvalidity][?<enc_search>]
```

```
imap://<iserver>/<enc_mailbox>[uidvalidity]<iuid>[isearch]
```

The first form is used to refer to an IMAP server, the second form refers to a list of mailboxes, the third form refers to the contents of a mailbox or a set of messages resulting from a search, and the final form refers to a specific message or message part. Note that the syntax here is informal. The authoritative formal syntax for IMAP URLs is defined in section 11."



## 6.4 InterfacelIdentifier

The InterfacelIdentifier object defines an instance of an interface that meets the appropriate element specifications in the object. The schema definition for this object follows:

```
<xsd:simpleType name="InterfacelIdentifierNameWKV">
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="JobControllInterface"/>
    <xsd:enumeration value="ServiceCapabilitiesInterface"/>
    <xsd:enumeration value="TargetDeviceSupportInterface"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="InterfacelIdentifierDataModelWKV">
  <xsd:restriction base="xsd:NMTOKEN">
    <xsd:enumeration value="pwg-sm"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:complexType name="InterfacelIdentifier">
  <xsd:all>
    <xsd:element name="name" type="intf:InterfacelIdentifierNameWKV"/>
    <xsd:element name="namespace" type="xsd:string"/>
    <xsd:element name="version" type="xsd:string"/>
    <xsd:element name="dataModel" type="intf:InterfacelIdentifierDataModelWKV"/>
    <xsd:element name="sec" type="xsd:string"/>
    <xsd:element name="auth" type="xsd:string"/>
  </xsd:all>
</xsd:complexType>

<xsd:element name="interfacelIdentifier" type="intf:InterfacelIdentifier"/>
```

The InterfacelIdentifier can be created by the client, and then passed to the server in the QueryEndPointsInterface.QueryInterfaceDefinition() method. This method checks to see if the service supports an interface that matches the criteria specified in the instance of the InterfacelIdentifier object.

Also, the QueryEndPointsInterface.QuerySupportedInterfaces() method returns a container of InterfacelIdentifiers that enumerate the interfaces that the service supports.

The InterfacelIdentifier object has the following elements:

name – The name of the interface

namespace – The namespace for the interface

version – The version of the interface

dataModel – The data model of the interface

sec: Well-known values defined in section 4.4.3 of [RFC2911] are:

'none': no security mechanism (i.e., 'anonymous').

'ssl3': SSL3 [SSL] security for communications channel.

'tls': TLS [RFC2246] security for communications channel.

auth: Well-known values defined in section 4.4.2 of [RFC2911] are:

'none': no user authentication (i.e., 'anonymous').

'login': user specified in login protocol operation. (Same as 'requesting-user-name' in [IPP])

'basic': HTTP basic authentication [RFC2617].

'digest': HTTP digest authentication [RFC2617].

'certificate': user authenticated via certificate.

## 7 Conformance

This section specifies the conformance requirements (necessary for basic interoperability) and conformance recommendations (useful for improved interoperability) for all implementations of the PSI/1.0 protocol. This section also specifies the conformance recommendations for PSI product literature (useful for informed customer purchasing decisions).

### 7.1 PSI/1.0 Common Conformance

#### 7.1.1 PSI/1.0 Common Implementation Requirements

The following common conformance REQUIREMENTS apply to every PSI/1.0 implementation (Client, Print Service, or Target Device).

Every PSI/1.0 implementation:

1. MUST support (produce and consume) valid [XML1.0] documents;
2. MUST support (produce and/or consume) valid [WSDL1.1] interfaces;
3. MUST support (produce and consume) valid [SOAP1.1] messages;
4. MUST support the [HTTP/1.1] binding of [SOAP/1.1];
5. MUST support the Printer Object and all mandatory elements, as defined in Section 7.6;
6. MUST support the Job Object and all mandatory elements, as defined in Section 7.7;
7. MUST support the Document Object and all mandatory elements, as defined in Section 7.8;
8. MUST support pwg-psips and pwg-psitd URL schemes for Target Devices, as defined in Section 6.2.3;

Rationale: Customers should be able to purchase PSI-based products with basic interoperability guaranteed for those PSI products.

#### 7.1.2 PSI/1.0 Common Implementation Recommendations

The following common conformance RECOMMENDATIONS apply to every PSI/1.0 implementation (Client, Print Service, or Target Device).

Every PSI/1.0 implementation:

1. SHOULD support [XML1.1] or later XML versions;

2. SHOULD support later WSDL versions;
3. SHOULD support [SOAP1.2] or later SOAP versions;
4. SHOULD support non-HTTP bindings (SMTP, BEEP, etc.) of [SOAP1.1] or later SOAP versions;
5. SHOULD support native device discovery for the implementation's supported environments (Windows, NetWare, UNIX, Linux, etc.);
6. SHOULD support secure sessions over [TLS1.0] or later TLS versions, as defined in Section 4.2;
7. SHOULD support the ipp URL scheme for Target Devices, as defined in Section 6.2.4;
8. SHOULD support the ftp URL scheme for References, as defined in Section 6.3.2.

Rationale: Customers should be able to purchase PSI-based products with improved interoperability probable for those PSI products.

### **7.1.3 PSI/1.0 Common Literature Recommendations**

The following common conformance RECOMMENDATIONS apply to product literature. All products that claim PSI/1.0 conformance:

1. SHOULD publish the role-independent checklist of conformance claims, as defined in applicable paragraphs of this Section 7.1 above, in installation, packaging, or other literature (e.g., Web pages);
2. SHOULD publish the role-specific checklist of conformance claims, as defined in applicable Sections 7.2, 7.3, and/or 7.4 below, in installation, packaging, or other literature (e.g., Web pages).

Rationale: Customers should be able to purchase PSI-based products with specific interoperability verified for those PSI products.

## **7.2 PSI/1.0 Client Conformance**

### **7.2.1 PSI/1.0 Client Requirements**

The following conformance REQUIREMENTS apply to a PSI/1.0 Client. Every PSI/1.0 Client implementation:

1. MUST support the client role in the JobControlInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.5.
2. MUST support the client role in the QueryEndpointsInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.3;

## 7.2.2 PSI/1.0 Client Recommendations

The following conformance RECOMMENDATIONS apply to a PSI/1.0 Client. Every PSI/1.0 Client implementation:

1. SHOULD support the client role in the ServiceCapabilitiesInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.4;
2. SHOULD support device discovery via the PSI Service Root URL as defined in Section 4.1
3. SHOULD support the http URL scheme for References, as defined in Section 6.3.1

## 7.3 PSI/1.0 Print Service Conformance

### 7.3.1 PSI/1.0 Print Service Requirements

The following conformance REQUIREMENTS apply to a PSI/1.0 Print Service. Every PSI/1.0 Print Service implementation:

1. MUST support the server role in the QueryEndpointsInterface, for access from PSI/1.0 Clients and PSI/1.0 Target Devices, as defined in Section 4.3;
2. MUST support the server role in the ServiceCapabilitiesInterface, for access from PSI/1.0 Clients and PSI/1.0 Target Devices, as defined in Section 4.4;
3. MUST support the server role in the JobControlInterface, for access from PSI/1.0 Clients and PSI/1.0 Target Devices, as defined in Section 4.5;
4. MUST support the server role in the TargetDeviceSupportInterface, for access from PSI/1.0 Target Devices, as defined in Section 4.6;
5. MUST support device discovery via the PSI Service Root URL as defined in Section 4.1.
6. MUST support the http URL scheme for References, as defined in Section 6.3.1.

### 7.3.2 PSI/1.0 Print Service Recommendations

The following conformance RECOMMENDATIONS apply to a PSI/1.0 Print Service. Every PSI/1.0 Print Service implementation:

1. SHOULD support the client role in the QueryEndpointsInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.3
2. SHOULD support the client role in the ServiceCapabilitiesInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.4;

3. SHOULD support the client role in the JobControlInterface, for access to PSI/1.0 Print Services and PSI/1.0 Target Devices, as defined in Section 4.5

## **7.4 PSI/1.0 Target Device Conformance**

### **7.4.1 PSI/1.0 Target Device Requirements**

The following conformance REQUIREMENTS apply to a PSI/1.0 Target Device. Every PSI/1.0 Target Device implementation:

1. MUST support the client role in the QueryEndpointsInterface, for access to PSI/1.0 Print Services, as defined in Section 4.3;
2. MUST support the client role in the JobControlInterface, for access to PSI/1.0 Print Services, as defined in Section 4.5;
3. MUST support the client role in the TargetDeviceSupportInterface, for access to PSI/1.0 Print Services, as defined in Section 4.6;
4. MUST support the server role in the QueryEndpointsInterface, for access from PSI/1.0 Clients and PSI/1.0 Print Services, as defined in Section 4.3;
5. MUST support the server role in the ServiceCapabilitiesInterface, for access from PSI/1.0 Clients and PSI/1.0 Print Services, as defined in Section 4.4;
6. MUST support the server role in the JobControlInterface, for access from PSI/1.0 Clients and PSI/1.0 Print Services, as defined in Section 4.5;
7. MUST support device discovery via the PSI Service Root URL as defined in Section 4.1.
8. MUST support the http URL scheme for References, as defined in Section 6.3.1.

### **7.4.2 PSI/1.0 Target Device Recommendations**

The following conformance RECOMMENDATIONS apply to a PSI/1.0 Target Device. Every PSI/1.0 Target Device implementation:

1. SHOULD support the client role in the ServiceCapabilitiesInterface, for access to PSI/1.0 Print Services, as defined in Section 5.4.

## 7.5 PSI/1.0 Method Support Requirements

Method	Print Service	Target Device	Client
QuerySupportedInterfaces	Mandatory to Implement	Mandatory to Implement	Optional to Use
QueryInterfaceDefinition	Mandatory to Implement	Mandatory to Implement	Mandatory to Use unless administratively pre-configured
GetTargetDeviceElements	Mandatory to Implement	Mandatory to Implement	Optional to Use
GetKnownTargetDevices	Mandatory to Implement	Mandatory to Implement	Optional to Use
ValidateReference	Mandatory to Implement	Mandatory to Implement	Optional to Use
CreateJob	Mandatory to Implement	Mandatory to Implement	Mandatory to Use
CloseJob	Mandatory to Implement	Mandatory to Implement	Mandatory to Use
AddDocumentByReference	Mandatory to Implement	Mandatory to Implement	Optional to Use (A Client shall call at least one AddDocument*() method.)
AddDocumentByPush	Mandatory to Implement	Mandatory to Implement	Optional to Use (A Client shall call at least one AddDocument*() method.)
PushDocumentDataDelivered	Mandatory to Implement	Mandatory to Implement	Conditionally Mandatory to Use after data is delivered for AddDocumentByPush operation.
AddDocumentByValue	Mandatory to Implement	Mandatory to Implement	Optional to Use (A Client shall call at least one AddDocument*() method.)
GetJobs	Mandatory to Implement	Mandatory to Implement	Optional to Use
GetJobElements	Mandatory to Implement	Mandatory to Implement	Optional to Use
SetJobElements	Mandatory to Implement	Mandatory to Implement	Optional to Use
CancelJob	Mandatory to Implement	Mandatory to Implement	Optional to Use
GetDocuments	Mandatory to Implement	Mandatory to Implement	Optional to Use
GetDocumentElements	Mandatory to Implement	Mandatory to Implement	Optional to Use
SetDocumentElements	Mandatory to Implement	Mandatory to Implement	Optional to Use
CancelDocument	Mandatory to Implement	Mandatory to Implement	Optional to Use
FetchDocumentDataByPull	Mandatory to Implement	Not Supported	Optional to Use
PullDocumentDataFetched	Mandatory to Implement	Not Supported	Optional to Use
FetchDocumentByValue	Mandatory to Implement	Not Supported	Optional to Use
FetchJobs	Recommended to Implement (necessary for firewall traversal)	Recommended to Implement (necessary for firewall traversal)	Optional to Use

FetchNextJob	Mandatory to Implement	Not Supported	Mandatory to Use
CloseJob	Mandatory to Implement	Not Supported	Mandatory to Use
FetchNextDocumentByPull	Mandatory to Implement	Not Supported	Conditionally Mandatory to Use (either FetchNextDocumentByPull or FetchNextDocumentByValue)
PullDocumentDataFetched	Mandatory to Implement	Not Supported	Mandatory to Use (Conditionally – after data is retrieved for FetchNextDocumentByPull operation.)
FetchNextDocumentByValue	Mandatory to Implement	Not Supported	Conditionally Mandatory to Use (either FetchNextDocumentByPull or FetchNextDocumentByValue)
SendJobNotification	Mandatory to Implement	Not Supported	Mandatory to Use if Print Service requests Job Notifications
SendDocumentNotification	Mandatory to Implement	Not Supported	Mandatory to Use
RegisterTargetDevice	Mandatory to Implement	Not Supported	Mandatory to Use before any other TargetDeviceSupportInterface method calls
UnregisterTargetDevice	Mandatory to Implement	Not Supported	Mandatory to Use
SendTargetDeviceNotification	Mandatory to Implement	Not Supported	Mandatory to Use

## 7.6 Printer Object Element Support Requirements

Listed elements are mandatory.

### 7.6.1 PrinterDescription

ColorSupported, DeviceID, DocumentFormatSupported, MultipleDocumentJobsSupported

### 7.6.2 PrinterStatus

All elements are mandatory

### 7.6.3 ProcessingDefaults, ProcessingReady, ProcessingSupported

No elements are mandatory

## 7.7 Job Object Element Support Requirements

Listed elements are mandatory

### 7.7.1 JobStatus Elements

JobURI, JobID, JobPrinterURI, JobState, JobStateReasons, NumberOfDocuments, TimeAtCompleted, TimeAtCreation, TimeAtProcessing, PrinterUpTime



### **7.7.2 JobDescription Elements**

JobName, JobOriginatingUserName

### **7.7.3 JobProcessing Elements**

No elements are mandatory

## **7.8 Document Object Element Support Requirements**

### **7.8.1 DocumentStatus Elements**

DocumentNumber, DocumentState, DocumentStateReasons, TimeAtCompleted, TimeAtCreation, TimeAtProcessing, PrinterUpTime

### **7.8.2 DocumentDescription Elements**

DocumentName, DocumentFormat, Compression

### **7.8.3 DocumentProcessing Elements**

No elements are mandatory

## **7.9 Job Lifetime**

A Print Service must keep a history of completed jobs for at least 5 minutes.

A Target Device must keep a history of completed jobs for at least 5 minutes.

If a Print Service that supports delayed Target Device association, then the Print Service shall wait for the Target Device to request the data for a minimum of 30 minutes.

If the client specifies a hold time, the service should abort the job after the hold time if the Target Device has not retrieved the job. (Note – this would allow a client to specify a hold time less than 30 minutes, then the Print Service shall honour the requested hold time.)

## 8 Normative References

[HTTP1.1]	Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and T. Berners-Lee, "Hypertext Transfer Protocol - HTTP/1.1", RFC 2616, June 1999.
[HTTPAuth]	Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A. and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.
[IANA-CHAR]	IANA Registry of Charset Tags, archived at: <a href="ftp://ftp.iana.org/assignments/character-sets">ftp://ftp.iana.org/assignments/character-sets</a>
[IANA-LANG]	IANA Registry of Language Tags, archived at: <a href="ftp://ftp.iana.org/assignments/language-tag-info">ftp://ftp.iana.org/assignments/language-tag-info</a>
[IANA-MIME]	IANA Registry of Media Types, archived at: <a href="ftp://ftp.iana.org/assignments/media-types">ftp://ftp.iana.org/assignments/media-types</a>
[IP]	Postel, J. "Internet Protocol", RFC 791, September 1981.
[IPSec]	Kent, S., Atkinson, R. "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
[RFC 2806]	Vaha-Sipila, A., "URLs for Telephone Calls", RFC 2806, April 2000.
[RFC 2817]	Khare, R., Lawrence, S., "Upgrading to TLS within HTTP/1.1", RFC 2817, May 2000.
[RFC 2910]	Herriot, R., Butler, S., Moore, P., Turner, R., Wenn, J., "Internet Printing Protocol/1.1: Encoding and Transport", RFC 2910, September 2000.
[PWG5105.1]	Candidate Standard PWG 5105.1-2004. Zehler, P., Hastings, T., Albright, S., "PWG Semantic Model", January 2004. (See <a href="http://www.pwg.org/sm/">http://www.pwg.org/sm/</a> ) <a href="ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf">ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf</a>
[SSL]	Defacto Industry Standard: Netscape, The SSL Protocol, Version 3, (Text version 3.02), November 1996.
[SOAP1.1]	Defacto Industry Standard: W3C Note "Simple Object Access Protocol (SOAP) 1.1", Don Box, David Ehnebuske, Gopal Kakivaya, Andrew Layman, Noah Mendelsohn, Henrik Nielsen, Satish Thatte, Dave Winer, 8 May 2000. (See <a href="http://www.w3.org/TR/SOAP/">http://www.w3.org/TR/SOAP/</a> )
[SOAP1.2]	[SOAP1.2-0], [SOAP1.2-1], and [SOAP1.2-2]
[SOAP1.2-0]	"SOAP Version 1.2 Part 0: Primer", W3C Recommendation, June 2003.
[SOAP1.2-1]	"SOAP Version 1.2 Part 1: Messaging Framework", W3C Recommendation, June 2003.
[SOAP1.2-2]	"SOAP Version 1.2 Part 2: Adjuncts", W3C Recommendation, June 2003.
[TCP]	Postel, J. "Transmission Control Protocol", RFC 793, September 1981.
[TLS]	Dierks, T. and C. Allen, "The TLS Protocol Version 1.0", RFC 2246, January 1999.
[URI]	Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1998.
[WSDL1.1]	Defacto Industry Standard: W3C Note "Web Services Description Language (WSDL) 1.1", Erik Christensen, Francisco Curbera, Greg Meredith, Sanjiva Weerawarana, 15 March 2001. (See <a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a> )
[XML]	See [XML1.0] and [XML1.1] below.
[XML1.0]	"Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation, October 2000.
[XML1.1]	"Extensible Markup Language (XML) 1.1", W3C Candidate Recommendation, October 2002.
[XSD]	[XSD-1] and [XSD-2] below.
[XSD-1]	"XML Schema Part 1: Structures", W3C Recommendation, May 2001.
[XSD-2]	"XML Schema Part 2: Datatypes", W3C Recommendation, May 2001.

## 9 Informative References

[SMB]	Hertel, C., "SMB Filesharing URL Scheme", July 13,2004. (See <a href="http://www.ietf.org/internet-drafts/draft-crhermel-smb-url-07.txt">http://www.ietf.org/internet-drafts/draft-crhermel-smb-url-07.txt</a> )
[ISO639]	See [ISO639-1] and [ISO639-2] below.
[ISO639-1]	"Codes for the Representation of Names of Languages -- Part 1: Alpha-2 Code", ISO/IEC 639-1, 2000.
[ISO639-2]	"Codes for the Representation of Names of Languages -- Part 2: Alpha-3 Code", ISO/IEC 639-2, 1998.
[ISO3166]	See [ISO3166-1] and [ISO3166-2] below.
[ISO3166-1]	"Codes for the Representation of Names of Countries and their Subdivisions, Part 1: Country Codes", ISO/ISO 3166- 1, 1997.
[PWG]	Printer Working Group, <a href="http://www.pwg.org">http://www.pwg.org</a>
[ISO3166-2]	"Codes for the Representation of Names of Countries and their Subdivisions, Part 2: Country Subdivision Codes", ISO/ISO 3166-2, 1998.
[RFC2046]	Freed, N., Borenstein, N. "MIME Part Two: Media Types", RFC 2046, November 1996.
[RFC2048]	Freed, N., Borenstein, N. "MIME Part Four: Registration Procedures", RFC 2048, November 1996.
[RFC2277]	Alvestrand, H. "IETF Policy on Character Sets and Languages", RFC 2277, January 1998.
[RFC2279]	Yergeau, F. "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.
[RFC2978]	Freed, N., Postel, J. "IANA Charset Registration Procedures", RFC 2978, October 2000.
[RFC3066]	Alvestrand, H. "Tags for the Identification of Languages", RFC 3066, January 2001.
[WSAddressing]	Bosworth,A., Box, D., Christensen, E., Cubera, F., Ferguson, D. Frey,J., Kaler, C., Langworthy, D., Leymann, F., Lucco, S., Millet, S., Mukhi, N., Nottingham, M., Orchard, D., Shewchuk, J., Storey, T., Weerawarana, S., "Web Services Addressing (WS-Addressing), March 2003.
[WSSecurity]	Atkinson, B., Della-Libera, G. Hada S., Hondo, M., Hallam-Baker, P., Klein, J., LaMacchia, B., Leach, P., Manferdeli, J., Maruyama, H., Nadalin, A., Nagaratnam, N., Prafullchandra, H., Shewchuk, J., Simon, D., "Web Services Security (WS-Security), April 2002.

## 10 PWG and IANA Considerations

This document does not require any new PWG or IANA registration support. The following paragraphs describe some existing PWG, IANA, W3C, and ISO registered standard elements used in PSI/1.0:

This PSI/1.0 document uses the existing standard elements of Job and Document objects defined in the PWG Semantic Model [PWG5105.1].

The accompanying PSI/1.0 WSDL references schema defined in [PWG5105.1].

The [XML] 'encoding' attribute contains a 'charset tag' [RFC2978]. New standard values may be registered according to [RFC2978] in the IANA Registry of Charset Tags [IANA-CHAR].

The [XML] 'lang' attribute contains a 'language tag' [RFC3066], i.e., a 'language code' [ISO639] optionally followed by a hyphen and a 'country code' [ISO3166]. New standard values (other than existing [ISO639] registrations) may be registered according to [RFC3066] in the IANA Registry of Language Tags [IANA-LANG].

The [PWG5105.1] 'DocumentNaturalLanguage' element also contains a 'language tag' [RFC3066].

The [PWG5105.1] 'DocumentFormat' element contains a MIME (content) media type [RFC2046]. New standard values may be registered according to [RFC2048] in the IANA Registry of Media Types [IANA-MIME].

## 11 Internationalization Considerations

PSI/1.0 implementations conform to all the best practice recommendations in "IETF Policy on Character Sets and Languages" [RFC2277].

PSI/1.0 implementations exchange [XML] information based on XML Schema [XSD] defined in the PWG Semantic Model [PWG5105.1].

The [XML] 'encoding' attribute (which MUST be supplied by PSI/1.0 Clients) SHOULD be set to 'utf-8' [RFC2279], as recommended by [RFC2277].

The [XML] 'lang' attribute (which SHOULD be supplied by PSI/1.0 Clients) SHOULD be set to a value conforming to [RFC3066], as recommended by [RFC2277].

The [PWG5105.1] 'DocumentNaturalLanguage' element (which SHOULD be supplied by PSI/1.0 Clients) SHOULD be set to a value conforming to [RFC3066], as recommended by [RFC2277].

## 12 Security Model and Considerations

This section describes the security threats against PSI/1.0 clients and servers. This section specifies the security conformance requirements and recommendations for PSI/1.0 implementations. This section addresses most security issues by reference to applicable underlying protocol specifications, for example, [SOAP1.1], [HTTP1.1] and [TLS].

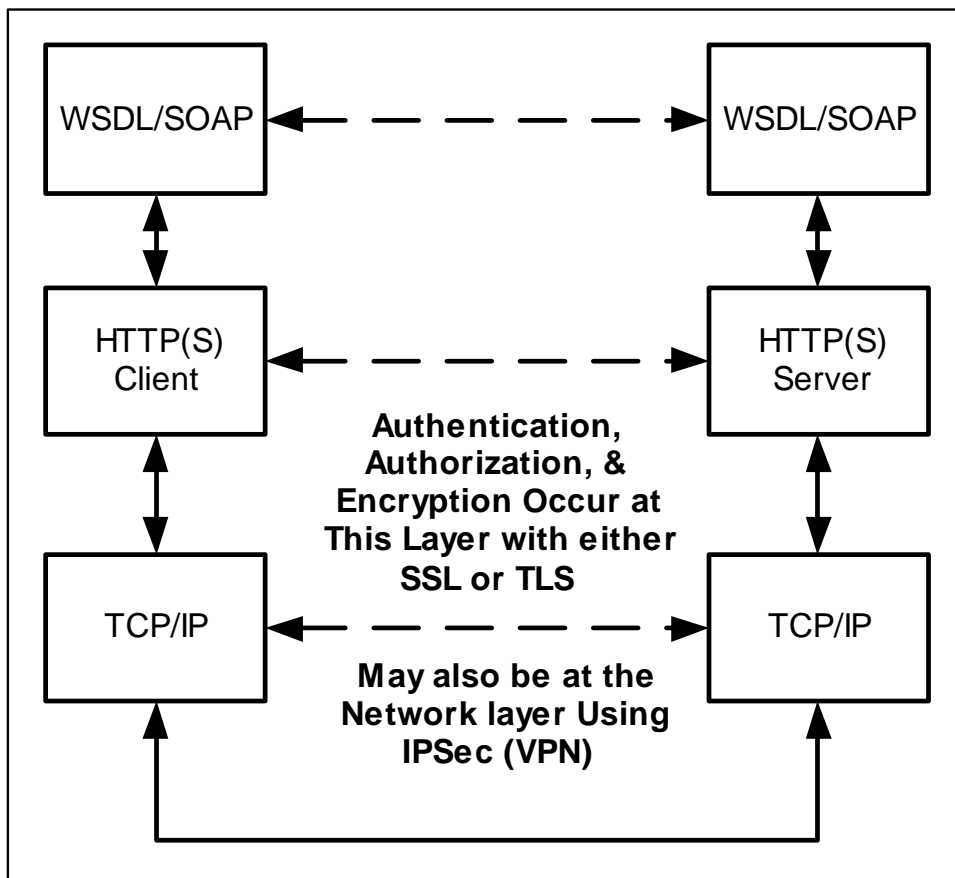


Figure 13 Security Model and Considerations

### 12.1 Internet Threat Model

This section is adapted from section 3 of "Guidelines for Writing RFC Text on Security Considerations" [RFC3552].

In the Internet threat model, we assume that the end-systems engaging in a protocol exchange have not themselves been compromised. Protecting against an attack when one of the end-systems has itself been compromised is extraordinarily difficult.

By contrast, we assume that the attacker has nearly complete control of the communications channel over which the end-systems communicate. This means that the attacker can read any PDU (Protocol Data Unit) on the network and undetectably remove, change, or inject forged packets onto the wire. This includes being able to generate packets that appear to be from a trusted machine. Thus, even if the end-system with which you wish to communicate is itself secure, the Internet environment provides no assurance that packets which claim to be from that system in fact are.

It's important to realize that the meaning of a PDU is different at different levels. At the [IP] level, a PDU means an IP packet. At the TCP level, it means a TCP segment. At the application layer, it means some kind of application PDU. For instance, at the level of [SOAP1.1], it means a single SOAP request or response message, while at the [HTTP1.1] level, it means a single HTTP request or response message.

### 12.1.1 Passive Attacks

In a passive attack, the attacker reads packets off the network but does not write them. The simplest way to mount such an attack is to simply be on the same LAN as the victim. On most common LAN configurations, including Ethernet, 802.3, and FDDI, any machine on the wire can read all traffic destined for any other machine on the same LAN. Note that switching hubs make this sort of sniffing substantially more difficult, since traffic destined for a machine only goes to the network segment that machine is located on.

Wireless communications channels deserve special consideration, especially with the recent and growing popularity of wireless-based LANs, such as those using 802.11. Since the data is simply broadcast on well known radio frequencies, an attacker simply needs to be able to receive those transmissions. Such channels are especially vulnerable to passive attacks. Although many such channels include cryptographic protection, it is often of very poor quality.

Passive attacks described in [RFC3552] and considered in PSI/1.0 are:

1. Confidentiality Violations - PSI/1.0 implementations MUST support [TLS] for protection against exposure of private business data.
2. Password Sniffing - PSI/1.0 implementations MUST support [TLS] and MUST NOT transfer cleartext passwords over unencrypted channels.
3. Offline Cryptographic Attacks - PSI/1.0 implementations MUST support [TLS] to protect against dictionary attacks against password-based challenge/response protocols [HTTPAuth].

### 12.1.2 Active Attacks

In an active attack, the attacker writes packets to the network and may read responses from the network. Active attacks that involve sending forged packets but not receiving any responses are called 'blind attacks'.

When IP [RFC791] is used without [IPSec], there is no authentication for the sender address. Active attacks that involve forging an IP packet with a false source address are called 'spoofing attacks'.

An IP packet with a forged source address may sometimes be screened out by the network infrastructure. For instance, many packet filtering firewalls screen out all packets with source addresses on the internal network that arrive on the external interface. Note that this provides no protection against an attacker who is inside the firewall. In general, protocol designers should assume that attackers can forge IP packets.

Not all active attacks require forging addresses. For example, the TCP SYN denial of service attack does not require disguising the sender's address. However, it is common practice to disguise one's address in order to conceal one's identity if an attack is discovered.

Active attacks described in [RFC3552] and considered in PSI/1.0 are:

1. Message Replay Attacks - PSI/1.0 implementations MUST support [TLS] and SHOULD always use at least [TLS] data integrity services for protection against [SOAP1.1] transaction replays by an attacker who has recorded a sequence of messages off the wire.
2. Message Insertion Attacks - PSI/1.0 implementations MUST support [TLS] and SHOULD always use at least [TLS] data integrity services for protection against message insertion by an attacker. PSI/1.0 implementations over [HTTP1.1] MUST take active measures to protect against denial-of-service attacks (for example, numerous incoming [TCP] connections from the same remote host or remote domain).
3. Message Deletion Attacks - PSI/1.0 implementations MUST support [TLS] and SHOULD always use at least [TLS] data integrity services for protection against message deletion by an attacker.
4. Message Modification Attacks - PSI/1.0 implementations MUST support [TLS] and SHOULD always use at least [TLS] data integrity services for protection against message modification by an attacker.
5. Man-In-The-Middle Attacks - PSI/1.0 implementations MUST support [TLS] and SHOULD always use at least [TLS] data integrity services and both Client and Server Authentication during [TLS] startup for protection against man-in-the-middle attacks.

## 12.2 Enterprise Threat Model

In the enterprise threat model, we can no longer assume that the end-systems engaging in a protocol exchange have not themselves been compromised. Physical security of workstations and network printers in an enterprise network is often the weakest point of security defenses. And print jobs typically produce hardcopy, which an inside attacker can simply steal.

Network security problems are actually worse in an enterprise network. Firewalls and border routers no longer provide any useful protection.

Users and administrators who deploy PSI/1.0 products in enterprise networks SHOULD enforce the use of [TLS] and SHOULD enforce the use of strong Client and Server Authentication during [TLS] startup.



## 12.3 Mobile Threat Model

In the mobile threat model, we can no longer defend against attackers based on network topology. Mobile clients may access home, business, or commercial PSI/1.0 products via:

1. Public Wireless - Cellular ISPs, IEEE 802.11 wireless Ethernet 'hot spots' in airports, etc.
2. Local Wireless - Bluetooth/IRDA-enabled laptops and printers, etc.

Users and administrators who deploy PSI/1.0 products in mobile networks SHOULD enforce the use of [TLS] and SHOULD enforce the use of strong Client and Server Authentication during [TLS] startup. [IPSec] also offers significant security advantages in mobile networks.

## 12.4 HTTP Threat Model

PSI/1.0 implementations are vulnerable to the following HTTP threats described in section 15 'Security Considerations' of [HTTP1.1]:

1. Attacks on personal information - [HTTP1.1] clients and servers in PSI/1.0 implementations MUST protect sensitive personal information, such as name, email address, etc. (see section 15.1 of [HTTP1.1]).
2. Attacks based on file and path names - [HTTP1.1] servers in PSI/1.0 implementations MUST NOT expose 'nearby' resources that were NOT explicitly configured for network access by administrators (see section 15.2 of [HTTP1.1]).
3. Attacks based on DNS spoofing - [HTTP1.1] clients and servers in PSI/1.0 implementations SHOULD NOT cache DNS name resolution results beyond their time-to-live value (see section 15.3 of [HTTP1.1]).
4. Attacks based on Location header spoofing - [HTTP1.1] servers in PSI/1.0 implementations MUST verify the validity of Location and Content-Location header data when supporting multiple trust domains (see section 15.4 of [HTTP1.1]).
5. Attacks based on Content-Disposition headers - [HTTP1.1] servers in PSI/1.0 implementations MUST defend against Content-Disposition header attacks (see section 15.5 of [HTTP1.1]).
6. Attacks based on retention of authentication credentials - [HTTP1.1] clients in PSI/1.0 implementations SHOULD NOT retain cached user authentication credentials beyond an administratively configured idle client time (see section 15.6 of [HTTP1.1]).
7. Attacks on HTTP Proxies - [HTTP1.1] servers in PSI/1.0 implementations SHOULD take active measures to defend against man-in-the-middle attacks and single source or distributed denial-of-service attacks (see section 15.7 of [HTTP1.1]).

## 13 Acronyms

HTTPS	HyperText Transfer Protocol – Secure
IP	Internet Protocol
IPSec	Internet Protocol SECurity
PDA	Personal Digital Assistant
PS	Print Service
PSI	Print Services Interface
SLP	Service Location Protocol
SMB	NetBios Application Protocol for File and Print Services
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
SSL	Secure Sockets Layer
TLS	Transaction Layer Security
UDDI	Universal Description, Discovery and Integration
UPNP	Universal Plug and Play
URL	Universal Resource Locator (see RFC2396)
WSDL	Web Service Description Language
XHTML	eXtensible HyperText Markup Language

## 14 Appendix A – Legacy URI schemes (Informative)

### 14.1 Target Device Identifier

#### 14.1.1 smb

smb should be of the form `smb://server/share` to facilitate use of existing UNC parsing code.

Query Examples:

```
smb://AparticularPrintServer/*?domain=AParticularDomain
```

```
smb:///*?name=Daves%20Printer
```

Result Examples:

```
smb://printserver1/printshare1?domain=AParticularDomain&name=Daves%20Printer&location=Daves%20Desk
```

```
smb://printserver1/printshare2
```

Specification Examples:

```
smb://printserver1/printshare3
```

Notes: See `draft-crhertel-smb-url-04.txt` for specification [SMB]

### 14.2 Reference

#### 14.2.1 smb:

Example:

```
smb://mydocumentserver/mydocumentshare/mydocument.txt
```

```
smb://svr/share/doc.txt?auth=login&user=dave&pw=davepw
```

## 15 Appendix B – Example XML Instance Objects (Informative)

### 15.1 unsupportedElements : UnsupportedElements

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XMLSPY v5 rel. 2 U (http://www.xmlspy.com)-->
<n:UnsupportedElements xmlns:n="http://www.pwg.org/schemas/sm/0.95/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pwg.org/schemas/sm/0.95/
C:\SEMANT~1\PwgObjects.xsd">
  <n:JobCopies>1</n:JobCopies>
</n:UnsupportedElements>
```

### 15.2 printerFilterElements : Printer

```
<?xml version="1.0" encoding="UTF-8"?>
<n:Printer xmlns:n="http://www.pwg.org/schemas/sm/0.95/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pwg.org/schemas/sm/0.95/
http://www.pwg.org/schemas/sm/0.95/PwgSemanticModel.xsd">
  <n:PrinterDescription>
    <n:ColorSupported>1</n:ColorSupported>
    <n:DocumentCharsetSupported>
      <n:DocumentCharset>iso-8859-1</n:DocumentCharset>
    </n:DocumentCharsetSupported>
    <n:DocumentFormatSupported>
      <n:DocumentFormat>application/octet-stream</n:DocumentFormat>
    </n:DocumentFormatSupported>
    <n:ImpressionsSupported>5</n:ImpressionsSupported>
    <n:JobPasswordEncryptionSupported>Md2</n:JobPasswordEncryptionSupported>
    <n:JobPasswordSupported>1</n:JobPasswordSupported>
    <n:JobSpoolingSupported>Automatic</n:JobSpoolingSupported>
    <n:MediaColDatabase>
      <n:MediaCol>
        <n:MediaHoleCount>3</n:MediaHoleCount>
        <n:MediaMaterial>aluminum</n:MediaMaterial>
      </n:MediaCol>
    </n:MediaColDatabase>
    <n:MultipleDocumentJobsSupported>1</n:MultipleDocumentJobsSupported>
  </n:PrinterDescription>
  <n:PrinterStatus>
    <n:PrinterIsAcceptingJobs>1</n:PrinterIsAcceptingJobs>
    <n:PrinterState>Idle</n:PrinterState>
  </n:PrinterStatus>
</n:Printer>
```

### 15.3 jobFilterElements : Job

```
<?xml version="1.0" encoding="UTF-8"?>
<n:Job xmlns:n="http://www.pwg.org/schemas/sm/0.95/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pwg.org/schemas/sm/0.95/
http://www.pwg.org/schemas/sm/0.95/PwgSemanticModel.xsd">
  <n:JobStatus>
    <n:JobId>0</n:JobId>
    <n:JobPrinterUri>smb://myprintserver/myprinter</n:JobPrinterUri>
  </n:JobStatus>
  <n:JobDescription>
    <n:JobAccountingID>2151232</n:JobAccountingID>
    <n:JobAccountingUserId>dhall@hp.com</n:JobAccountingUserId>
    <n:JobOriginatingUserName>David Hall</n:JobOriginatingUserName>
    <n:JobPassword>A9FD64E12C</n:JobPassword>
  </n:JobDescription>
</n:Job>
```

```
<n:JobPasswordEncryption>Md2</n:JobPasswordEncryption>
</n:JobDescription>
</n:Job>
```

## 15.4 documentFilterElements : Document

```
<?xml version="1.0" encoding="UTF-8"?>
<n:Document xmlns:n="http://www.pwg.org/schemas/sm/0.95/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.pwg.org/schemas/sm/0.95/
http://www.pwg.org/schemas/sm/0.95/PwgSemanticModel.xsd">
  <n:DocumentStatus>
    <n:JobId>0</n:JobId>
  </n:DocumentStatus>
  <n:DocumentDescription>
    <n:DocumentFormat>application/octet-stream</n:DocumentFormat>
  </n:DocumentDescription>
</n:Document>
```

## 16 Appendix C – Standard Print System Events (Normative)

### 16.1 Introduction

This appendix defines the names and semantics for the PWG Standard Print System Events. These standard events may occur on various managed components of Print Systems (e.g., Channels, Jobs, Documents, etc.). Print Systems may deliver these standard events to subscribed network management stations or other network host systems in notifications.

#### 16.1.1 Rationale for Defining Standard Events

Network printers are rapidly evolving into more powerful network print systems. Most new network printers include IPP/1.1 [RFC2911], the core source document for the PWG Semantic Model/1.0 [PWG5105.1]. Meanwhile, remote fleet management of network printers is becoming increasingly common. This document defines the PWG Standard Print System Events/1.0 in order to foster industry coherence in the monitoring and management of network print systems.

#### 16.1.2 Informative References in Standard Events

At the end of most standard event definitions in this appendix there are one or more `_informative_` references to:

- `<Reference>`
- - related operations, appropriate for queries after notifications, e.g.,
- See: `Get-Job-Attributes` in section 3.3.4 of IPP/1.1 [RFC2911].
- - related attributes, appropriate for inclusion in notifications, e.g.,
- See: `document-state` in section 9.1.25 of [PWG5100.5].

Note: This appendix does NOT define the standard content of either subscriptions or notifications.

#### 16.1.3 Naming of Document Events

This appendix defines the names and semantics of the Document events by analogy to the equivalent Job events.

#### 16.1.4 Naming of Job Events

This appendix defines the names and semantics of the Job events `_consistently_` with the IETF IPP Event Notifications and Subscriptions [IPP-NOT].

Note: This appendix does NOT normatively depend on [IPP-NOT].

#### 16.1.5 Naming of Printer Events

This appendix defines the names and semantics of the Printer events `_consistently_` with the IETF IPP Event Notifications and Subscriptions) [IPP-NOT].

Note: This appendix does NOT normatively depend on [IPP-NOT] for these Printer event definitions.

### 16.1.6 Naming of Subunit Events

This appendix defines the names and semantics of the Subunit events *\_consistently\_* with the `prtAlertCode` columnar object in Printer MIB v1 [RFC1759] and the `PrtAlertCodeTC` textual convention in IANA Printer TC [RFC3805].

Note: This appendix does NOT normatively depend on [RFC1759] or [RFC3805] for these Subunit event definitions.

## 16.2 Terminology

### 16.2.1 Model Terminology

This appendix imports (normatively) the definitions and semantics of the following model terms from the referenced documents:

- <Model Term>
- Component - see Printer MIB v1 [RFC1759]
- Document - see PWG Semantic Model/1.0 [PWG5105.1]
- Job - see PWG Semantic Model/1.0 [PWG5105.1]
- Printer - see PWG Semantic Model/1.0 [PWG5105.1]
- Subunit - see Printer MIB v1 [RFC1759]
  - - Input Tray (subunit)
  - - Interpreter (subunit)
  - - Marker (subunit)
  - - Marker Supplies (subunit)
  - - Media Path (subunit)
  - - Output Bin (subunit)

Note: The PWG Semantic Model/1.0 [PWG5105.1] in turn imports Printer and Job object semantics from IPP/1.1 Model and Semantics [RFC2911] and Document object semantics from the IPP Document Object [PWG5100.5].

### 16.2.2 Conformance Terminology

This appendix imports (normatively) the definitions and semantics of the following conformance terms from IETF Keywords for Use in RFCs to Indicate Requirement Levels [RFC2119]:

- <Conformance Term>
- MUST
- MUST NOT
- REQUIRED
- SHOULD
- SHOULD NOT
- RECOMMENDED

- MAY
- OPTIONAL

## 16.3 Standard Document Events

This appendix defines the names and semantics of the Document events by analogy to the equivalent Job events.

### 16.3.1 DocumentStateChanged

The Document state or state reasons have changed (i.e., the value of the DocumentState or DocumentStateReasons elements has changed).

This standard event has the following more specific sub-events: DocumentStateOnlyChanged, DocumentCreated, and DocumentCompleted.

- See: document-state in section 9.1.25 of [PWG5100.5].
- See: document-state-reasons in section 9.1.27 of [PWG5100.5].

#### 16.3.1.1 DocumentStateOnlyChanged

The Document basic state has changed (i.e., the value of the DocumentState element has changed). State reason changes do not cause this event.

- See: DocumentStateChanged above.
- See: document-state in section 9.1.25 of [PWG5100.5].

#### 16.3.1.2 DocumentCreated

A new Document has been created.

- See: DocumentStateChanged above.
- See: Print-Job in section 3.2.1 of IPP/1.1 [RFC2911].
- See: Print-URI in section 3.2.2 of IPP/1.1 [RFC2911].
- See: Send-Document in section 3.3.1 of IPP/1.1 [RFC2911].
- See: Send-URI in section 3.3.2 of IPP/1.1 [RFC2911].

#### 16.3.1.3 DocumentCompleted

The Document has terminated (i.e., the value of the DocumentState element has changed to 'Canceled', 'Aborted', or 'Completed').

- See: DocumentStateChanged above.
- See: Cancel-Document in section 4.5 of [PWG5100.5].

### 16.3.2 DocumentConfigChanged

The Document configuration has changed (i.e., the value of one or more non-state Document elements has changed).

- See: Set-Document-Attributes in section 4.4 of [PWG5100.5].
- See: Get-Document-Attributes in section 4.3 of [PWG5100.5].

### 16.3.3 DocumentProgress

The Document has completed processing one more media sheet (for hardcopy jobs) or image (for softcopy jobs).

This standard event has the following more specific sub-events: DocumentError and DocumentWarning.



See: k-octets-processed in sect 9.1.34 of [PWG5100.5].  
 See: impressions-completed in sect 9.1.31 of [PWG5100.5].  
 See: impressions-completed-current-copy in 9.1.32 of [PWG5100.5].  
 See: media-sheets-completed in section 9.1.37 of [PWG5100.5].  
 See: job-collation-type in section 4.1 of [RFC3381].  
 See: sheet-completed-copy-number in section 4.2 of [RFC3381].  
 See: sheet-completed-document-number in section 4.3 of [RFC3381].  
 See: impressions-completed-current-copy in sect 4.4 of [RFC3381].

### 16.3.3.1 DocumentError

The Document has encountered a processing error.

See: 'errors-detected' value of document-state-reasons  
 in section 9.1.27 of [PWG5100.5].  
 See: errors-count in section 9.1.29 of [PWG5100.5].

### 16.3.3.2 DocumentWarning

The Document has encountered a processing warning.

See: 'warnings-detected' value of document-state-reasons  
 in section 9.1.27 of [PWG5100.5].  
 See: warnings-count in section 9.1.45 of [PWG5100.5].

## 16.4 Standard Job Events

This appendix defines the names and semantics of the Job events consistently with the IETF IPP Event Notifications and Subscriptions [IPP-NOT].

### 16.4.1 JobStateChanged

The Job state or state reasons have changed (i.e., the value of the JobState or JobStateReasons elements has changed).

This standard event has the following more specific sub-events: JobStateOnlyChanged, JobCreated, JobCompleted, and JobStopped.

See: job-state in section 4.3.7 of IPP/1.1 [RFC2911].  
 See: job-state-reasons in section 4.3.8 of IPP/1.1 [RFC2911].

#### 16.4.1.1 JobStateOnlyChanged

The Job basic state has changed (i.e., the value of the JobState element has changed). State reason changes do not cause this event.

See: JobStateChanged above.  
 See: job-state in section 4.3.7 of IPP/1.1 [RFC2911].

#### 16.4.1.2 JobCreated

A new Job has been created.

See: JobStateChanged above.  
 See: Print-Job in section 3.2.1 of IPP/1.1 [RFC2911].  
 See: Print-URI in section 3.2.2 of IPP/1.1 [RFC2911].  
 See: Create-Job in section 3.2.4 of IPP/1.1 [RFC2911].

### 16.4.1.3 JobCompleted

The Job has terminated (i.e., the value of the JobState element has changed to 'Canceled', 'Aborted', or 'Completed').

See: JobStateChanged above.

See: Cancel-Job in section 3.3.3 of IPP/1.1 [RFC2911].

### 16.4.1.4 JobStopped

The Job has stopped (i.e., the value of the JobState element has changed to 'ProcessingStopped'). Human intervention may be required to correct the fault condition(s) and resume Job processing.

See: JobStateChanged above.

## 16.4.2 JobConfigChanged

The Job configuration has changed (i.e., the value of one or more non-state Job elements has changed).

See: Set-Job-Attributes in section 4.2 of [RFC3380].

See: Get-Job-Attributes in section 3.3.4 of IPP/1.1 [RFC2911].

## 16.4.3 JobProgress

The Job has completed processing one more media sheet (for hardcopy jobs) or image (for softcopy jobs).

This standard event has the following more specific sub-events: JobError and JobWarning.

See: job-k-octets-processed in section 4.3.18.1 of [RFC2911].

See: job-impressions-completed in section 4.3.18.2 of [RFC2911].

See: job-media-sheets-completed in section 4.3.18.3 of [RFC2911].

See: job-collation-type in section 4.1 of [RFC3381].

See: sheet-completed-copy-number in section 4.2 of [RFC3381].

See: sheet-completed-document-number in section 4.3 of [RFC3381].

See: impressions-completed-current-copy in sect 4.4 of [RFC3381].

### 16.4.3.1 JobError

The Job has encountered a processing error.

See: 'job-completed-with-errors' value of job-state-reasons in section 4.3.8 of IPP/1.1 [RFC2911].

See: errors-count in section 5.1.1 of IPP Job Ext [PWG5100.7].

### 16.4.3.2 JobWarning

The Job has encountered a processing warning.

See: 'job-completed-with-warnings' value of job-state-reasons in section 4.3.8 of IPP/1.1 [RFC2911].

See: warnings-count in section 5.1.4 of IPP Job Ext [PWG5100.7].

## 16.5 Standard Printer Events

This appendix defines the names and semantics of the Printer events consistently with the IETF IPP Event Notifications and Subscriptions [IPP-NOT].

### 16.5.1 PrinterStateChanged

The Printer state or state reasons have changed (i.e., the value of the PrinterState, PrinterStateReasons, or PrinterIsAcceptingJobs elements has changed).

This standard event has the following more specific sub-events: `PrinterStateOnlyChanged`, `PrinterRestarted`, `PrinterShutdown`, and `PrinterStopped`.

See: `printer-state` in section 4.4.11 of IPP/1.1 [RFC2911].  
See: `printer-state-reasons` in section 4.4.12 of IPP/1.1 [RFC2911].  
See: `printer-is-accepting-jobs` in section 4.4.23 of IPP/1.1 [RFC2911].

#### 16.5.1.1 *PrinterStateOnlyChanged*

The Printer basic state has changed (i.e., the value of the `PrinterState` element has changed). State reason changes do not cause this event.

See: `PrinterStateChanged` above.  
See: `printer-state` in section 4.4.11 of IPP/1.1 [RFC2911].

#### 16.5.1.2 *PrinterRestarted*

The Printer has been completely restarted (with or without a power cycle).

See: `PrinterStateChanged` above.

#### 16.5.1.3 *PrinterShutdown*

The Printer has been completely shut down (i.e., disabled from accepting any new incoming Jobs, paused from processing any existing Jobs, and finally terminated as a network service).

See: `PrinterStateChanged` above.

#### 16.5.1.4 *PrinterStopped*

The Printer has stopped (i.e., the value of the `PrinterState` element has changed to 'Stopped'). Human intervention may be required to correct the fault condition(s) and resume Printer processing.

See: `PrinterStateChanged` above.  
See: `printer-state` in section 4.4.11 of IPP/1.1 [RFC2911].  
See: `printer-state-reasons` in section 4.4.12 of IPP/1.1 [RFC2911].

### 16.5.2 **PrinterConfigChanged**

The Printer configuration has changed (i.e., the value of one or more non-state/non-queue Printer elements has changed).

This standard event has the following more specific sub-events: `PrinterMediaChanged` and `PrinterFinishingsChanged`.

See: `Set-Printer-Attributes` in section 4.1 of [RFC3380].  
See: `Get-Printer-Attributes` in section 3.2.5 of IPP/1.1 [RFC2911].  
See: `xxx-[default|ready|supported]` in IPP/1.1 [RFC2911].  
See: `pvtGeneralConfigChanges` in [RFC1759] and [RFC3805].

#### 16.5.2.1 *PrinterMediaChanged*

The hardcopy input media currently loaded on this Printer has changed (i.e., the value of the `MediaReady` element has changed).

See: `PrinterConfigChanged` above.  
See: `media-[default|ready|supported]` in IPP/1.1 [RFC2911].  
See: `pvtInputDefaultIndex` in [RFC1759] and [RFC3805].

### 16.5.2.2 PrinterFinishingsChanged

The finishings currently available on this Printer have changed (i.e., the value of the FinishingsReady element has changed).

See: PrinterConfigChanged above.  
See: finishings- [default|ready|supported] in IPP/1.1 [RFC2911].  
See: prtOutputDefaultIndex in [RFC1759] and [RFC3805].

### 16.5.3 PrinterQueueOrderChanged

The Job queue order on this Printer has changed (i.e., the value of the NumberOfInterveningJobs element has changed in one or more Jobs).

See: Get-Jobs in section 3.2.6 of IPP/1.1 [RFC2911].  
See: job-priority in section 4.2.1 of IPP/1.1 [RFC2911].  
See: number-of-intervening-jobs in section 4.3.12 of IPP/1.1 [RFC2911].

## 16.6 Standard Subunit Events

This appendix defines the names and semantics of the Subunit events consistently with the prtAlertCode columnar object in Printer MIB v1 [RFC1759] and the PrtAlertCodeTC textual convention in IANA Printer TC [RFC3805].

Note: This appendix does NOT normatively depend on [RFC1759] or [RFC3805] for these Subunit event definitions.

### 16.6.1 Alert Management

AlertRemovalOfBinaryChangeEntry

### 16.6.2 General (Print System)

The following self-explanatory Subunit events apply only to an overall network Print System:

GeneralCoverOpen  
GeneralCoverClosed  
GeneralInterlockOpen  
GeneralInterlockClosed  
GeneralConfigurationChange  
GeneralJam  
GeneralPowerUp  
GeneralPowerDown  
GeneralManagementReset  
GeneralManualReset  
GeneralReadyToProcess

### 16.6.3 Input Tray

The following self-explanatory Subunit events apply only to an Input Tray component of a Print System:

InputMediaTrayMissing  
InputMediaSizeChange  
InputMediaWeightChange  
InputMediaTypeChange  
InputMediaColorChange  
InputMediaFormPartsChange  
InputMediaSupplyLow  
InputMediaSupplyEmpty  
InputMediaChangeRequest  
InputManualInputRequest

InputTrayPositionFailure  
 InputTrayElevationFailure  
 InputCannotFeedSizeSelected

#### 16.6.4 Interpreter

The following self-explanatory Subunit events apply only to an Interpreter component of a Print System:

InterpreterMemoryIncrease  
 InterpreterMemoryDecrease  
 InterpreterCartridgeAdded  
 InterpreterCartridgeDeleted  
 InterpreterResourceAdded  
 InterpreterResourceDeleted  
 InterpreterResourceUnavailable  
 InterpreterComplexPageEncountered

#### 16.6.5 Marker

The following self-explanatory Subunit events apply only to a Marker component of a Print System:

MarkerFuserUnderTemperature  
 MarkerFuserOverTemperature  
 MarkerFuserTimingFailure  
 MarkerFuserThermistorFailure  
 MarkerAdjustingPrintQuality

#### 16.6.6 Marker Supplies

The following self-explanatory Subunit events apply only to a Marker (or Marker Supplies) component of a Print System:

MarkerTonerEmpty  
 MarkerInkEmpty  
 MarkerPrintRibbonEmpty  
 MarkerTonerAlmostEmpty  
 MarkerInkAlmostEmpty  
 MarkerPrintRibbonAlmostEmpty  
 MarkerWasteTonerReceptacleAlmostFull  
 MarkerWasteInkReceptacleAlmostFull  
 MarkerWasteTonerReceptacleFull  
 MarkerWasteInkReceptacleFull  
 MarkerOpcLifeAlmostOver  
 MarkerOpcLifeOver  
 MarkerDeveloperAlmostEmpty  
 MarkerDeveloperEmpty  
 MarkerTonerCartridgeMissing

#### 16.6.7 Media Path

The following self-explanatory Subunit events apply only to a Media Path component of a Print System:

MediaPathMediaTrayMissing  
 MediaPathMediaTrayAlmostFull  
 MediaPathMediaTrayFull  
 MediaPathCannotDuplexMediaSelected

#### 16.6.8 Output Bin

The following self-explanatory Subunit events apply only to an Output Bin component of a Print System:

OutputMediaTrayMissing

```
OutputMediaTrayAlmostFull
OutputMediaTrayFull
OutputMailboxSelectFailure
```

### 16.6.9 Subunit (Generic)

The following self-explanatory Subunit events apply to any component of a Print System:

```
SubunitMissing
SubunitLifeAlmostOver
SubunitLifeOver
SubunitAlmostEmpty
SubunitEmpty
SubunitAlmostFull
SubunitFull
SubunitNearLimit
SubunitAtLimit
SubunitOpened
SubunitClosed
SubunitTurnedOn
SubunitTurnedOff
SubunitOffline
SubunitPowerSaver
SubunitWarmingUp
SubunitAdded
SubunitRemoved
SubunitResourceAdded
SubunitResourceRemoved
SubunitRecoverableFailure
SubunitUnrecoverableFailure
SubunitRecoverableStorageError
SubunitUnrecoverableStorageError
SubunitMotorFailure
SubunitMemoryExhausted
SubunitUnderTemperature
SubunitOverTemperature
SubunitTimingFailure
SubunitThermistorFailure
```

## 16.7 Vendor Events

Conformance: Vendors SHOULD name their extension events according to the following ABNF [RFC2234] pattern (including the vendor prefix).

```
vndextension    = vndprefix "-" vndname
vndprefix      = "vnd" 1*(LOWERALPHA)
vndname       = (object / subunit) name
object        = "Printer" / "Job" / "Document"
subunit       = "Alert" / "General" / "Input" /
               "Interpreter" / "Marker" / "MediaPath" /
               "Output" / "Subunit"
name          = 1*(word)
word          = 1(UPPERALPHA) *(LOWERALPHA / DIGIT)
DIGIT        = %x30-39           ; 0-9 (decimal digit)
UPPERALPHA   = %x41-5A         ; A-Z (uppercase alpha)
LOWERALPHA   = %x61-7A         ; a-z (lowercase alpha)
```

For example:

```
vndacme-PrinterButteredToast      -- a clever printer
```

Conformance: Vendors SHOULD NOT define extensions to the standard Subunit events, for improved interoperability.

Note: The vendor prefix does NOT use a colon separator (used in many namespace systems) in order to avoid conflicts with existing protocols that do not allow colons in enumerated attribute values.

## 16.8 Best Practices for Future Events Standards

Future public standards that address events in Print Systems should import and use the standard events defined in this appendix and should reference a specific PWG-approved version of this document.

Future public standards may also case normalize and/or insert hyphens or underscores between words in these standard event names, according to other naming conventions.

Future public standards should not make any other changes to the standard event names defined in this appendix, for improved interoperability.

## 16.9 Normative References (for this appendix)

[PWG5105.1] Candidate Standard PWG 5105.1-2004. Zehler, P., Hastings, T., Albright, S., "PWG Semantic Model", January 2004. (See <http://www.pwg.org/sm/>) <ftp://ftp.pwg.org/pub/pwg/candidates/cs-sm10-20040120-5105.1.pdf>

[RFC1759] Smith, Wright, Hastings, Zilles, Gyllenskog. Printer MIB, RFC 1759, March 1995.

[RFC2119] Bradner. Keywords for Use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997.

[RFC2234] Crocker, Overell. Augmented BNF for Syntax Specifications (ABNF), IETF RFC 2234, November 1997.

## 16.10 Informative References (for this appendix)

[RFC3805] Bergman, Lewis, McDonald. Printer MIB v2, RFC3805, July 2004.

[PWG5100.5] Hastings, Zehler, et al. Internet Printing Protocol (IPP): Document Object, PWG 5100.5, October 2003.

[PWG5100.7] Hastings, Zehler, et al. Internet Printing Protocol (IPP): Job Extensions, PWG 5100.7, October 2003.

[RFC2911] Hastings, Herriot, deBry, Isaacson, Powell. Internet Printing Protocol/1.1: Model and Semantics, RFC 2911, September 2000.

[RFC3380] Hastings, Herriot, Kugler, Lewis. Internet Printing Protocol (IPP): Job and Printer Set Operations, RFC 3380, September 2002.

[RFC3381] Hastings, Lewis, Bergman. Internet Printing Protocol (IPP): Job Progress Attributes, RFC 3381, September 2002.

[IPP-NOT] Harriott, Hastings. IESG Approved Internet Printing Protocol: Event Notifications and Subscriptions, <draft-ietf-ipp-not-spec-12.txt>, June 2004